A Study of Key Management and Location Management for Mobile Networks
(モバイルネットワークにおける鍵管理および位置管理に関する研究)

September 2005

Jun ANZAI

Principal Advisor
Professor Tsutomu Matsumoto

Graduate School of Environment and Information Sciences
Yokohama National University

Acknowledgements

I would like to express my gratitude to my advisor Professor Tsutomu Matsumoto for leading my research. In addition, I am grateful to Associate professor Junji Shikata for beneficial advices. Also, I want to thank all the professors Seiichiro Kagei, Tomoharu Nagao and Tatsunori Mori for their useful comments.

I would like to thank Mr. Naohiko Noguchi and other members of secure mobile development group of Panasonic Mobile Communications Co., Ltd. I would like to thank Mr. Noritake Okada and Mr. Satoshi Terasaki because they give me a wonderful opportunity to study as a doctorial course student at Yokohama National University. I greatly thank Mr. Takehiko Kato and Dr. Natsume Matsuzaki since loaning to AMSL (Advanced Mobile Telecommunications Security Technology Research Laboratories Co., Ltd.).

I thank all members of Professor Tatsutomu Matsumoto's laboratory (especially, Dr. Kenichiro Akai, Dr. Katsunari Yoshioka, Dr. Masataka Suzuki, Mr. Manabu Nakano, and Mr. Tsukasa Kayama) and the staff of the Department of Information Media and Environment Sciences of Yokohama National University for their support while I was attending Yokohama National University.

Finally, I want to express my deepest gratitude to my wife Chikako, my daughter Tsugumi, and my son Tenga for their support and encouragemet.

Abstract

The subjects of this thesis are secure key management and secure location management for mobile nodes on wireless mobile networks.

Recently, mobile networks become increasingly popular with the widespread availability of cellular phones and notebook PC's. Moreover new mobile networks are appearing, for example autonomous and flexible networks as typified by ad-hoc / mesh networks and ubiquitous computing / networks as typified by sensor networks and radio frequency identification (RFID) systems.

On wireless mobile networks, attackers are easy to eavesdrop and connect communication channels. Therefore it is important to prevent attacks in comparison to traditional wired networks. Cryptography is an essential tool in information security. Symmetric cryptography uses the same key (symmetric key) for message encryption and message decryption. In case that a valid user Alice has a symmetric key secretly, Alice can securely share a secret message with another user Bob whom Alice selects by giving the symmetric key to Bob secretly. Also, asymmetric cryptography uses a key (public key) to encrypt a secret message and uses another key (private key) to decrypt an encrypted secret message. According to publish a public key, the corresponding private key holder can decrypt encrypted messages that other users encrypt using the public key. In general, public key infrastructure (PKI) that a trusted third party certificates an identity of a public key is used for knowing whose public key. Such techniques (symmetric key sharing and PKI) are called "key management". Thus cryptography requires secure key management. Here, mobile networks have significant features that are different from traditional wired networks, and these features are 1) dynamic change of network topology, 2) no authorities and so on. Consequently, there are still many key management problems.

On the other hand, many mobile nodes are always moving on mobile networks. Therefore it is important that network managers and service providers obtain location information of mobile nodes. As location measurement technology, there are techniques that calculate a location using a difference time between radio waves as typified by global positioning system (GPS) and techniques that confirm a location using information, which can be obtained in a specific location, as typified by RFID and beacons. Newly, location-based services (LBSs), which depend on geographical locations of mobile nodes, receive mach attention on ubiquitous computing. Services of LBSs include content distribution systems to a specific location, navigation systems for walkers, tracking systems of mobile nodes and so on. In the near future, this thesis expects to increase LBSs that require high security and anonymity. Therefore secure location management technology is major subject of study.

Moreover the thesis considers cryptographic LBSs that are comprised mainly of key management and location management techniques for enhancing safety.

In Chapter 1 describes overview of this thesis and the contribution.

In Chapter 2, this thesis defines a system model consists of common properties abstracted from some target networks. One of the properties is that mobile nodes are not always connected to stations, which are gates of backbone infrastructure such as base stations of cellular phone services and access points of wireless LAN services. Thus this thesis assumes a proxy node manages keys and locations of other nodes instead of a station.

In Chapter 3, this thesis indicates key management problems on the system model and proposes new schemes for solving the problems. In case that a service provider requires to treat many mobile nodes as one group for providing services, the nodes should share a temporally group key. On the other hand, it is important that many honest nodes, which forward messages of other nodes and support PKI-processing, exist for maintenance of ad-hoc / mesh networks. From the above-mentioned instances, network managers should urge cooperation to mobile nodes. Here, the target networks have the following problems:

1. A dynamic group key sharing demands heavy calculation costs to nodes because the nodes are not always connected to stations that can assist in node calculations.
2. In case of occurring trouble, group members cannot certificate members who share a group key and a time when sharing the group key to a third party because of sharing the key without stations.
3. It is difficult to force nodes to cooperate because of not using enforceable stations.

The thesis proposes new schemes that solve the above problems by using a proxy node instead of a station.

In Chapter 4, this thesis indicates location management problems on the target networks and proposes secure location verification schemes for solving the problems. Location verification schemes aim to solve *Location Verification Problem* (a verifier V verifies the fact that a prover P exists in a location L at a time T). For solving *Location Verification Problem*, plural location verification schemes are proposed, however the schemes have the following problems:

1. The schemes are not secure against *Relay Attack*.
2. The schemes cannot verify relation distances between plural provers.

Relay Attack is that a dishonest P can force V to believe a fake location L' by a relay station exists between P and V. The thesis shows that the attack can be applied to the existing schemes using communication delay and proposes new schemes resistant against the attack. Also, the existing schemes cannot verify relation distances between provers unless plural verifiers exist. The thesis proposes plural provers verifiable schemes by extending the scheme against resistant *Relay Attack*.

In Chapter 5, the thesis defines cryptographic LBSs. For example, cryptographic LBSs are a system in which a user can read a secret business document stored in a notebook PC in an office

however re-encryption of the document prevents its reading when the notebook PC is removed from the office, and a system in which a user can use a digital sign key (as an official company seal) stored in a device for a specific room. In addition, the thesis indicates that cryptographic LBSs are insecure in due to incomplete integration of key management and location management functions and proposes a secure method of combining these functions.

Finally, in Chapter 6, this thesis summarizes the thesis and shows future works as follows:

1. for realizing the proposed location verification schemes, and
2. for verifying validity of the proposed secure cryptographic LBSs constructing method.

Contents

Chapter 1. Introduction

1. 1. Background

Mobile networks become increasingly popular a part of modern life, with the widespread availability of cellular phones and notebook PC's. Moreover, new mobile networks are appearing, for example autonomous and flexible networks as typified by ad-hoc / mesh networks and ubiquitous computing / networks as typified by sensor networks and radio frequency identification (RFID) systems.

On wireless mobile networks, attackers are easy to eavesdrop and connect communication channels. Therefore it is important to prevent attacks in comparison to traditional wired networks. Cryptography is an essential tool in information security because the cryptography can protect digitized information. Cryptography is classified into symmetric cryptography and asymmetric cryptography. Symmetric cryptography uses the same key (symmetric key) for message encryption and message decryption. Also, asymmetric cryptography uses a key (public key) to encrypt a secret message and uses another key (private key) to decrypt an encrypted secret message. Namely, cryptography uses information gap between with or without keys. A valid user Alice has a symmetric key that can decrypt an encrypted message. Alice can share a secret message with another user Bob whom Alice selects by giving the symmetric key to Bob.

On the other hand, a public key can be published, since the public key cannot decrypt an encrypted message. In addition, it is hard to obtain the corresponding private key from the public key. A private key holder can decrypt encrypted messages that any users encrypt using the corresponding public key. However, these users do not want to use the public key if they cannot verify an identifier of the public key. In general, users use public key infrastructure (PKI) that a trusted third party certificates an identity of a public key.

Such techniques (key sharing and PKI) are called "key management". Key management is one of most important techniques in cryptography. Here, mobile networks have significant features that are different from traditional wired networks, and these features are 1) dynamic change of network topology, 2) no authorities, and so on. Consequently, there are still many key management problems.

On the other hand, many mobile nodes are always moving on wireless mobile networks. Therefore it important that network managers and service providers obtain locations of target mobile

nodes. Heretofore, it has already realized to location measurement techniques using global positioning system (GPS) and radar. Location measurement technology is a method that measures geographic locations of objects. As location measurement technology, there are techniques that calculate a location using a difference time between radio waves as typified by GPS and radar, and techniques that confirm a location using information, which can be obtained in a specific location, as typified by sensor networks, RFID, and beacons. On security of these technologies, researchers have studied a method to verify validity of locations, and a method to protect privacy information (i.e. location information) of mobile nodes.

Newly, context awareness services, which use contexts of mobile nodes, are studied actively on ubiquitous computing / networks. Especially, location-based services (LBSs), which use geographical location information of mobile nodes as contexts, receive mach attention. Services of LBSs include information distribution systems to a specific location, navigation systems for walkers, tracking systems of mobile nodes, and location-based access control systems (e.g. a ticket gate), along with other applications. With diversification of LBSs in the near future, this thesis expects to increase LBSs that require high security and anonymity. Therefore secure location management technology is major subject of study.

1. 2. Contribution of the Thesis

The subjects of this thesis are secure management for keys and locations of mobile nodes on wireless mobile networks. At first the thesis discusses key management and location management on mobile networks individually. Finally the thesis defines cryptographic LBSs that are comprised mainly of these management techniques for enhancing security, and considers an integration method of key management and location management functions for realizing secure cryptographic LBSs.

At first the thesis indicates three problems of key management techniques on the target networks and systems, and then proposes three schemes for solving the problems. A service provider requires treating a set of mobile nodes as one group for providing services. Therefore the nodes should share a temporally group key. Also, it is important that many honest nodes, which forward messages of other nodes and support PKI-processing, exist for maintenance of networks, because a mobile node doubles with a router in ad-hoc / mesh networks. From the above-mentioned instances, network managers should urge cooperation to mobile nodes. Here, the target networks have the following three problems:

1. A dynamic group key sharing demands heavy calculation costs to low performance nodes, because the nodes are not always connected to high performance stations that can assist in node calculations on the target systems. Here, the stations are gates of backbone infrastructure such as base stations of cellular phone services and access points of wireless LAN services.
2. In case of occurring trouble, group members cannot certificate members who share a group key and a time when sharing the group key to a third party, because of sharing the group key without stations.
3. It is difficult to force nodes to cooperate because of not using enforceable stations.

The thesis proposes new schemes that solve the above problems by assuming a proxy node instead of a station.

Secondly the thesis indicates two problems of existing location management techniques using communication delay on the target networks, and then proposes two secure location verification schemes for solving the problems. Location verification schemes aim to solve *Location Verification Problem* (a verifier V verifies the fact that a prover P exists in a location L at a time T). For solving *Location Verification Problem*, plural location verification schemes have been proposed. However the schemes have the following problems:

The existing schemes are not secure against *Relay Attack*.

The existing schemes cannot verify relation distances between plural provers.

Relay Attack is that a dishonest P can force V to believe a fake location L' of P if a relay station *RS* operated by P exists between P and V. The thesis shows that the attack can be applied to the existing location verification schemes using communication delay, and then proposes a scheme resistant against the attack. On the other hand, the existing schemes can verify a distance between plural provers and a verifier however cannot verify relation distances between plural provers unless plural verifiers exist. The thesis shows plural provers verifiable location verification schemes can be constructed by extending the scheme against resistant *Relay Attack*.

Finally, the thesis defines cryptographic LBSs by which LBSs use cryptography for enhancing security. In addition, the thesis considers security of cryptographic LBSs, and then proposes a method of constructing secure cryptographic LBSs. Instances of cryptographic LBSs are a system in which a user can read a secret business document stored in a notebook PC in an office however re-encryption of the document prevents its reading when the notebook PC is removed from the office, and a system in which a user can use a digital sign key (as an official company seal) stored in a device for a specific room. On the other hand, the thesis indicates that cryptographic LBSs are insecure in due to incomplete integration of key management and location management functions, for example a user would like to share a key with a mobile node of a specific location, however an attacker who exists on other location forces the user to share the key with the attacker. The thesis proposes a secure method of integrating key management and location management functions for preventing the above attack. Also the thesis suggests a potential for new cryptographic LBSs by showing plural combinations of key management and location management functions.

1. 3. Outline of the Thesis

Figure 1.3.1 shows organization of this thesis. The thesis is organized as follows.

In Chapter 2, the thesis defines a system model consists of nodes, proxy nodes, stations, and a backbone infrastructure.

In Chapter 3, Section 3.1 indicates three problems of existing key management technologies on the system model. Next, the thesis proposes three schemes: 1) a distributed user revocation scheme, 2) interaction key generation schemes, and 3) an integrated the incentive and the PKI-supporting mechanism, as solutions of the problems to Section 3.2, Section 3.3, and Section 3.4, respectively.

In Chapter 4, Section 4.1 indicates two problems of existing location verification schemes using communication delay. Then, the thesis proposes two schemes: 1) a location verification scheme resistant against relay attack, and 2) plural provers verifiable location verification schemes, as solutions of the problems to Section 4.2, and Section 4.3, respectively.

In Chapter 5, Section 5.1 defines cryptographic LBSs and shows examples of cryptographic LBSs. In Section 5.2, the thesis considers security of cryptographic LBSs, and proposes a method of constructing secure cryptographic LBSs.

In Chapter 6, Section 6.1 summarizes the thesis. Section 6.2 describes future works.

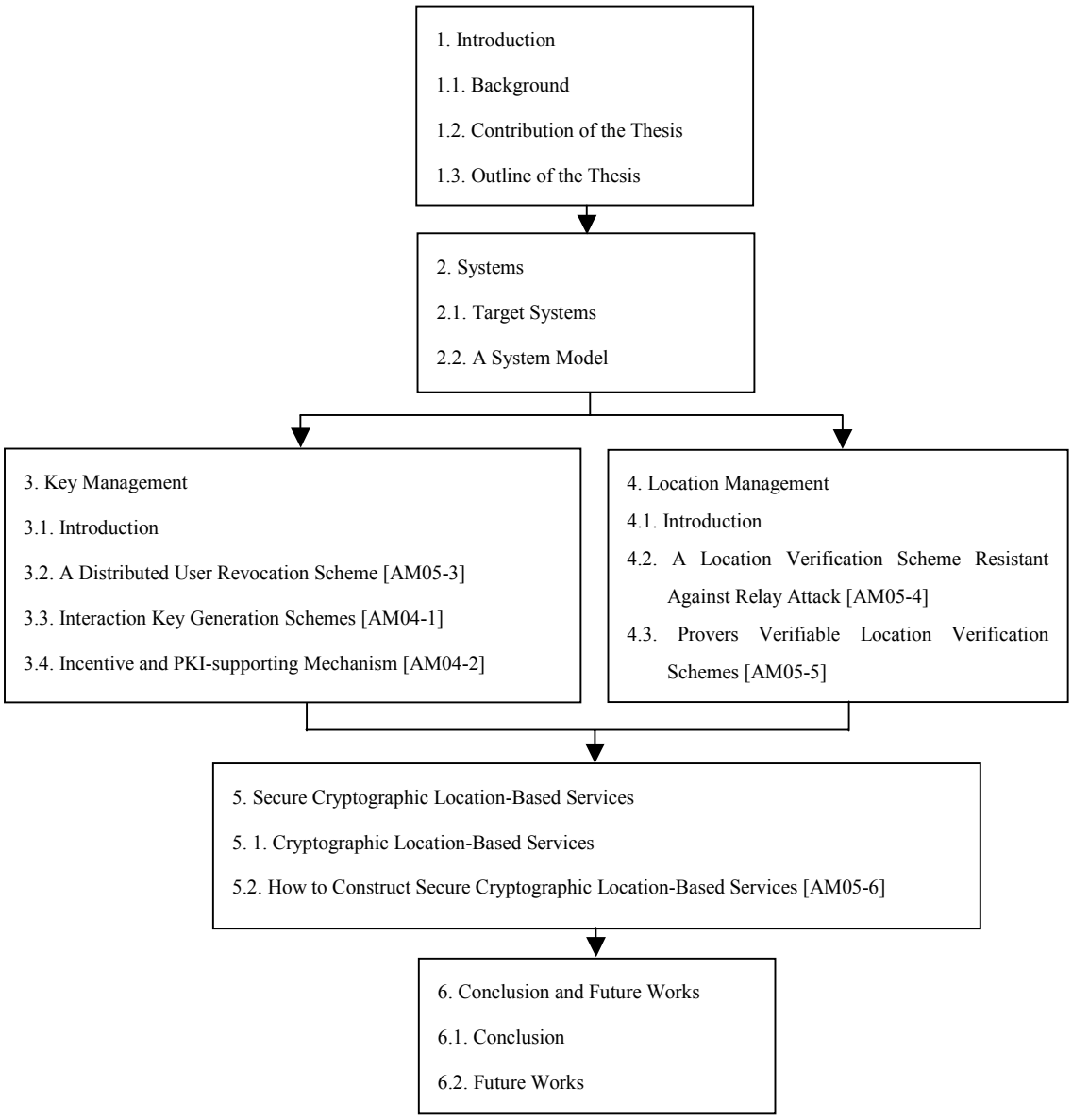


Fig. 1.3.1. Construction of the Thesis

Chapter 2. Systems

2. 1. Target Systems

Target systems of this thesis are the following wireless systems and combinations of those systems:

Ad-hoc / mesh networks: are composed of a set of autonomous mobile nodes. The nodes multi-hop communicate with each other node without backbone infrastructures.

Cellular phone service systems: are composed of cellular phones and base stations. The cellular phones connect to backbone infrastructures via the base stations.

Wireless LAN spot service systems: are composed of wireless LAN terminals (e.g. a PDA and a notebook PC) and access points. The terminals connect to backbone infrastructures via the access points.

Multi-hop cellular networks: are integrated networks of a cellular network (e.g. a cellular phone system and a wireless LAN spot service system) and an ad-hoc network.

Sensor networks: are composed of a set of autonomous fixed sensor nodes and a data collection center. The sensor nodes multi-hop communicate with each other node without backbone infrastructures. The sensor nodes send sensed data to the data collection center.

Radio frequency identification (RFID) systems: are IC tags, IC tag read / writers, and databases. The IC tag read / writer reads an ID from the IC tag, and then the read / writer sends the ID to the database. The tags and read / writers are either fixed or mobile.

The above wireless communication systems are studied actively now because the systems relate to ubiquitous computing / networks. The thesis presumes that ubiquitous computing / networks include one of the above systems or combinations of those systems. Therefore, the thesis also target ubiquitous computing / networks. Ubiquitous computing / networks provide context awareness services that depend on contexts of mobile nodes. Especially, a location-based service (LBS), which uses geographical location information of mobile nodes as context, receives mach attention. Services of LBSs include information distribution to a specific location, navigation for walkers, tracking of mobile nodes, and location-based access control (e.g. a ticket gate), along with other applications.

Wireless multi-hop communication is used for ad-hoc / mesh networks, multi-hop cellular networks, and sensor networks. In addition, wireless LAN spot service systems may use wireless multi-hop communication. Wireless multi-hop communication is that one or more relay nodes relay messages from a sending node to a receiving node as bucket brigade.

2. 2. A System Model

In this section, this thesis defines a system model consists of the following entities:

A **backbone infrastructure** is an authority that manages stations and wire-communicates with stations. The corresponding communication channel is secure.

A **station** is a gate of a backbone infrastructure, for instance a base station, an access point, a database, and a data collection center. A station wire-communicates with other stations via a backbone infrastructure and wireless-communicates with nodes. A station has high-performance and may provide LBSs to nodes.

A **node** is a wireless mobile terminal that can wireless-communicate with stations and other nodes. However a node cannot directly communicate with a backbone infrastructure. Also a node has low or middle performance and may be a dishonest entity.

The system model abstracts the following common properties from the target systems.

Wireless mobile communication

Wireless communication channel is insecure.

Either there are no a backbone infrastructure and a station, or it is not always possible for a node to connect to a station.

The thesis focuses on the above third property since the property is an essential problem of mobile networks. Therefore, the thesis assumes the following proxy node for solving the problem.

A **proxy node** is a node that has middle-performance. A proxy node is not authority. However a proxy node is trustworthier than a general node. Because a proxy node is temporally authorized by stations, or is elected by general nodes. In addition, a proxy node may provide LBSs to a general node.

Figure 2.2.1 shows an instance of the system model. On the figure, node A is a proxy node and arrows means multi-hop communication.

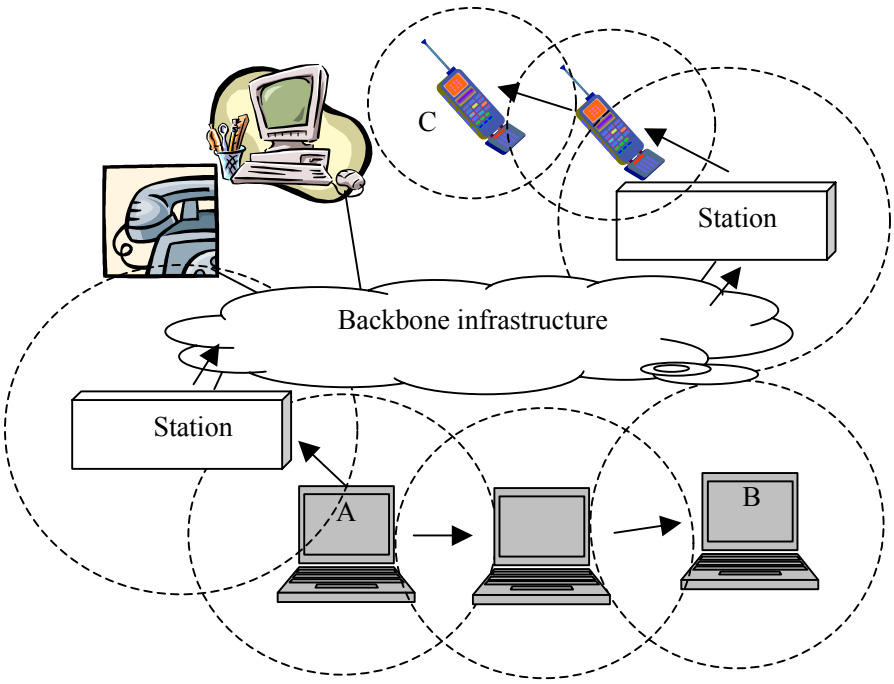


Fig. 2.2.1. The system model

Chapter 3. Key Management

3. 1. Introduction

On wireless mobile networks, attackers are easy to eavesdrop and connect communication channels. Therefore it is important to prevent attacks in comparison to traditional wired networks. Cryptography is an essential tool in information security because cryptography can protect digitized information against attacks. Cryptography is also a technology using information gap between with or without keys, thus keys must be stored secretly. Here, mobile networks have features that are different from traditional wired networks. Those features are 1) dynamic change of network topology, 2) no authorities and so on. Consequently, there are still many key management problems.

This section indicates three key management problems on the target networks and systems, and then proposes three schemes for solving the three problems.

A service provider requires to treat a set of mobile nodes as one group for providing services, therefore the nodes should share a temporally group key. On the other hand, it is important that many honest nodes, which forward messages of other nodes and support PKI-processing, exist for maintenance of networks, because a node doubles with a router in ad-hoc / mesh networks. Thus network managers (e.g. stations) should urge cooperation to nodes. From the above-mentioned instances, the target networks and systems have the following three problems:

1. A dynamic group key sharing demands heavy calculation costs to low-performance nodes because the nodes are not always connected to high-performance stations on the system model.
2. In case of occurring trouble, group members cannot certificate members who share a group key and a time when sharing the group key, because of sharing the group key without stations.
3. It is hard to force nodes to cooperate because of not using enforceable stations.

By assuming a proxy node instead of a station, this thesis proposes new schemes, which are 1) a distributed user revocation scheme, 2) interaction key generation schemes, and 3) an integrated the incentive and the PKI-supporting mechanism, as solutions of the problems to Section 3.2, Section 3.3, and Section 3.4, respectively.

Section 3.2 describes a user revocation scheme for decentralized networks. User revocation is a method of transmitting a group key shared by n users so that all but d-revoked users can obtain the group key. On decentralized networks such as ad-hoc / mesh networks and Peer-to-Peer (P2P) networks, a sender should revoke the access of a dishonest user or an unauthorized user as soon as possible to protect the security of group communication. However, it would take a long time to

revoke a user in a large group if the sender distributes the group key to all users aside from the revoked user. In addition, users must set shared group keys for each user without a privileged center. The thesis proposes a scheme in which the amount of transmission and the key storage of each user are small.

Section 3.3 describes a new concept of "*Interaction key*". An interaction key is a group public key that corresponds to a shared private key, which is shared by multiple users. An interaction key has a new feature that an interaction key generator can verify the following facts: the shared private key has been generated now, and the shared private key has not existed before. In other words, the multiple users can prove those facts to the key generator. This feature is different from time-stamp technologies prove that a message existed at a point in time. Here, the key generator is a third party that can observe communications of the multiple users. Existing technologies only allow a group member or a privileged entity to generate a group public key. The thesis is not presently aware of a technology where a third party can generate the group public key as above. For example, the interaction key technology is useful both for generating public key certificates and for message certification. On certificate generation, a certificate authority (i.e. an interaction key generator) can issue a public key certificate for an interaction key. On message certification, the users can prove the signed message has not existed before, because the message is signed by the shared private key corresponding to the interaction key.

Section 3.4 describes a new scheme that integrates between an incentive mechanism and a PKI-supporting mechanism. Multi-hop communication networks (e.g. ad-hoc / mesh networks and multi-hoc cellular networks) have two problems: 1) PKI is not always available, and 2) nodes do not always cooperate to forward messages. This section indicates a relation between the problems, and proposes a new security mechanism that can unitedly solve the problems.

3. 2. A Distributed User Revocation Scheme

3. 2. 1. Introduction

3. 2. 1. 1. Background

Recently, decentralized networks such as ad-hoc networks, mesh networks, and Peer-to-Peer (P2P) networks, are receiving much attention. In decentralized networks, users of devices such as PC's, PDAs, and cellular telephones can communicate mutually without a privileged center (e.g. a base station or a trusted server). Moreover, users demand secure group communication without dependence on the center.

This study specifically addresses ad-hoc networks. Ad-hoc networks are multi-hop networks formed by a set of mobile nodes in a self-organizing manner without relying on any fixed infrastructure. The nodes must perform all networking functions (e.g., routing, packets forwarding, etc.) because ad-hoc networks have no infrastructure including routers, centers, base stations, access points, and so on.

In general, *broadcast encryption* is used for secure group communication. Broadcast encryption allows a sender to send a message to all authorized users simultaneously and privately over a broadcast channel. Pay-TV and Video On Demand (VOD) are typical examples of systems that can use broadcast encryption. In those systems, a secure and fast method is desired to distribute a shared key (a *group key*) to all authorized users. However, most broadcast encryption schemes that presume to use a privileged center are unsuitable for ad-hoc networks. Therefore, users must set each shared group key for each user without the center.

This paper specifically presents *user revocation*, which is a kind of broadcast encryption. User revocation is a method of transmitting a group key shared by n users so that all but d-revoked users can obtain the group key. A sender should revoke users as quickly as possible. In addition, large costs are incurred when revoking users in a large group if the sender distributes the group key to all users aside from the revoked user(s).

On the other hand, the Diffie-Hellman key exchange scheme and its variants are famous as means of conference key sharing. Some studies have expanded the Diffie-Hellman scheme to methods that share a key among three or more entities; others have applied the Diffie-Hellman scheme to ad-hoc networks.

The study proposes a scheme in which the amount of transmissions and key storage of respective users are acceptably small. Moreover, the proposed scheme requires no privileged center for user revocation and initialization that a phase sets user's secret key into each user. This study particularly

focuses on ad-hoc networks.

3. 2. 1. 2. Related Works

One simple method to revoke d (< n) users from n users is that a sender distributes a new group key to each user, aside from the d revoked user(s), as an encrypted form using a secret key of each user. This method requires each user to retain only one secret key, whereas the sender should transmit n - d encrypted new group keys. Another simple method is that each user has common keys for every subset of n users. This method requires no sender to transmit any message, whereas each user should keep many keys.

A first major step occurred when two research groups of Wallner et al. [WHA99] and Wong et al. [WGL97] proposed a key distribution scheme using a logical key hierarchy (called LKH scheme in this paper). In the LKH schemes, the amount of transmissions is $O((degree - 1) \cdot log\ n)$ and the number of keys for each user is $O(log\ n)$, where n is the number of users in a group, and *degree* is the number of users in a bottom subgroup of the logical key hierarchy. However, the LKH schemes require a privileged center for setting user keys for each user.

A second major step occurred when two works on user revocation were presented: one by Anzai et al. [AMM99-4], and one by Naor et al. [NP00]. This thesis calls their schemes Revocation-based on Secret Sharing (RSS) in this paper. The RSS scheme by Anzai et al. uses a (k, $n + k$ -1) threshold secret sharing scheme with a Diffie-Hellman key exchange scheme. The scheme requires a user to store one shared secret (called a shadow in this paper); a sender distributes k messages containing shared secrets for revoking d ($\leq k$) users. The scheme has a property by which any user can become a sender. Naor et al. independently proposed similar RSS scheme [NP00]. The scheme considers traceability, but the sender is a fixed and privileged center. Unfortunately, these RSS schemes require the center for setting user's secret key (i.e. a shadow) into each user.

Recently, new trends exist: a combination of a LKH scheme and an RSS scheme, and a group key distribution scheme that is suitable for decentralized networks. Combination scheme [KMSW01] of the RSS scheme by Anzai et al. and LKH scheme proposed by Kurnio et al. By giving shadows hierarchically to each user, the combination scheme [KMSW01] can incur less transmission cost of revoking users than the transmission cost of the RSS scheme, but each user must have as many shadows as the LKH scheme. For ad-hoc networks, Cho and Kim proposed a combination scheme [CK00] of the Pedersen scheme [P91] and a hierarchy structure of LKH scheme. The combination scheme requires no privileged center for user revocation and initialization. Moreover, when a user leaves a group, the user transmits only one notification to the group. However, other users of the group are obligated to remake their own shadow after each session; consequently, each user must keep all parameters to share a secret. In addition, it is necessary to communicate via proxy nodes of subgroups when a sender sends an encrypted message to all members, because all subgroups do not

have the same group key.

As a group key distribution scheme suitable for P2P networks, scheme [NW04] proposed by Watanabe et al. is appreciated in a previous paper [WKSEYY03]. (Note that the present paper does not refer to the scheme directly. Nevertheless, the scheme presented herein is a newer version of a scheme explained in that paper.) That scheme, which allows Pedersen scheme to be more flexible, requires no privileged center for user revocation and initialization. However, each user must communicate with other users for every user revocation. In addition, as a group key-sharing scheme that is suitable for ad-hoc networks, the scheme [STW00] proposed by Steiner et al. is appreciated in this paper [AD02]. The scheme expands the Diffie-Hellman key exchange scheme to share a key among three or more entities through a ring network and a broadcast network. For group key sharing, the scheme repeats that a user relays its own key information and a previous user's information to a subsequent user; then a final user broadcasts all modified key information to a group. Therefore, the last user, who knows all key information, is a privileged user. Using a ring network, the scheme offers an advantage: the transmission costs are smaller than that of the well-known extended Diffie-Hellman scheme [BD94].

Table 3.2.1 shows that all the above-mentioned schemes have at least one of three problems.

Table 3.2.1. Related works each present at least one of three problems.

Problems	Schemes
User revocation and initialization require a privileged center	[AMM99-4][KMSW01][NP00] [STW00][WGL97][WHA99]
Sending encrypted messages depends on transmission routes	[CK00]
Transmissions and key storage costs for user revocation are great	[CK00][NW04]

3. 2. 1. 3. Our Goal

This paper proposes a scheme that satisfies the following four requirements:

Decentralization: a privileged center is not required for initialization and user revocation.

Independence: sending encrypted messages does not depend on transmission routes. A sender can send messages directly to all members.

Efficiency: costs of revoking users and initialization are sufficiently small.

Resistance: the scheme has security against collusion of more than *d*-revoked users. The collusion threshold should not allow any combinations of users because numerous possible

combinations exist on ad-hoc networks.

Through the use of these features, the proposed scheme is applicable to decentralized networks. The proposed scheme is especially suitable for wireless ad-hoc networks because multi-hop communications and broadcast communications can reduce the transmission costs of the proposed scheme.

To achieve our goal, this study applies the Pedersen scheme to the RSS scheme by Anzai et al. because the Pedersen scheme provides decentralization to the RSS scheme. (A similar idea is described roughly in another paper [DHS03].) Subsequently, the study adopts a concept of multistage secret sharing [S79][TO98] for initialization because only the first approach is unable to increase efficiency for initialization. The second approach reduces costs of the Pedersen scheme and gives independence to the proposed scheme because our multistage approach allows some group members to use the Pedersen scheme. In contrast, scheme [CK00] hierarchically repeats the Pedersen scheme for multistaging. In this manner, the costs of this multistage approach are greater than that of our multistage approach. The scheme [CK00] cannot obtain independence because of the breaking off of relations of shadows among subgroups.

Moreover, the second approach improves security, since a secret leaks out at particular combinations of shadows if the number of leaked shadows reaches a threshold. On the other hand, the Pedersen scheme leaks a secret value at any combination of leaked shadows. According to their approaches, the study expects that all the requirements can be satisfied. In addition, the study evaluates the calculation costs of a reduction technique [AMM99-3] for the RSS scheme. Then this study shows that the technique is the most suitable for the proposed scheme.

The proposed scheme requires public key cryptography. Therefore, this study explains ways to apply public key infrastructure (PKI) to ad-hoc networks. A simple method is that each user has a public key certificate (with the corresponding private key) issued by a certificate authority and signs public keys of the proposed scheme using the private key. In addition, the proposed scheme can use PKI schemes [CBH02][KZLLZ01][YK02][ZH99] for ad-hoc networks.

3. 2. 2. The Proposed Scheme

This section presents the proposed scheme. The proposed scheme assumes mobile nodes as users and can be based on an appropriate *Diffie-Hellman Problem* defined over finite cyclic groups, including subgroups of Jacobians of elliptic curves, and so on. This section explains the proposed scheme over a prime field $\boldsymbol{Z}_p$.

A group comprising all nodes has some low-stage subgroups. A low-stage subgroup must contain a proxy node. In addition, all proxy nodes form a high-stage subgroup. For that reason, the proposed scheme allows the power of a privileged center to distribute to plural proxy nodes.

The scheme contains two phases: an initialization phase and a node revocation phase. After explaining the target system and assumptions, the thesis describes the two phases in 3.2.2.2 and 3.2.2.3 respectively.

3. 2. 2. 1. A Target System and Assumptions

A target system comprises the following:

Node *i*: A node (i.e. user) labeled i is identified with its ID number as a group member; the node belongs to at least one low-stage subgroup. The study assumes that the total number of nodes in the target system is n and the total number of nodes in a low-stage subgroup x is n_x. Let Δ_x be a set of ID numbers in a low-stage subgroup x. Let Γ_x be a set of spare low ID numbers in a low-stage subgroup x. Here, the spare low ID numbers are used when the number of revoked nodes is less then k_x - 1, where k_x is a threshold value of Shamir's secret sharing scheme [S79] on a low-stage subgroup x.

Proxy node *x*: A proxy node is a node that is elected by other nodes for initialization; she manages her own low-stage subgroup. This study assumes that the total number of all proxy nodes and spare high ID numbers is t ($\leq n$). Let Φ be a set of ID numbers of all proxy nodes. Let Γ be a set of spare high ID numbers. Here, the spare high ID numbers are used when the number of revoked proxy nodes is less then k - 1, where k is a threshold value of Shamir scheme. In an initialization phase, the proxy nodes initialize all shadows (i.e. secret keys) for all nodes. In the proposed scheme, any node can become a proxy node when elected.

Revoked node *j*: A node (or proxy node) is revoked from a group. Let Λ be a set of ID numbers of revoked proxy nodes. The set has d proxy nodes. Let Λ_x be a set of ID numbers of revoked nodes in a low-stage subgroup x, having d_x nodes.

Sender: A sender is an entity that can send an encrypted message to a group. The sender determines a revoked node(s), and sends revocation data that is used to revoke the revoked node. The study assumes that a sender can access a public bulletin board.

In addition, the public bulletin board maintains system parameters and public keys (or information to calculate the public keys) for all nodes. Two methods to realize the board exist. A first method is that proxy nodes share the board. A second method is that a server (e.g. station) keeps the board. This server is not a privileged center because it does not have only one secret.

The study makes the following security assumptions:

1. A communication channel is not secure: anyone can obtain data on the channel.
2. The *Diffie-Hellman Problem* is computationally hard to solve.
3. In (k, $n + k$ - 1) threshold cryptosystems, anyone with less than k shadows cannot get information about a secret S.
4. Revoked nodes may conspire to get a group key.

5. Revoked nodes may publish their secret information to damage system security.
6. Valid nodes do not conspire with revoked nodes. A revoked node cannot obtain a group key from a valid node if this assumption is valid.
7. Valid nodes do not publish their secret keys.
8. A public bulletin board has each parameter with the corresponding certificates. It shall be checked before using the public parameters.
9. The following system parameters are published to the public bulletin board:

   p: a large prime number,

   q: a large prime number such that $q \mid p - 1$), and

   g: a q^{th} root of unity over $\boldsymbol{Z}_p$.

3. 2. 2. 2. Initialization Phase

At the beginning, an initialization phase is carried out only one time.

Generation of a Proxy node's Shadow.

This procedure is similar to a Pedersen scheme key generation.

1. A proxy node x decides a parameter k satisfying

   $$0 \le d \le k - 2 < t.$$

2. The proxy node x generates a secret random number r_x ($\in \boldsymbol{Z}_q$), and stores it secretly.
3. The proxy node x generates random integers $a_{x,0}, a_{x,1}, a_{x,2}, \ldots, a_{x,k-1}$ that satisfy the following conditions:

   $$a_{x,0} = rx,\ 0 \le a_{x,w} \le q - 1 \text{ for all } 1 \le w \le k - 1, \text{ and } a_{x,k-1} \ne 0.$$

4. The proxy node x calculates as

   $$F_x = \{F_{x,w} (= g^{a_{x,w}} \bmod p) \mid \text{for } w = 0, 1, 2, \ldots, k - 1\}.$$

5. The proxy node x defines the following equation.

   $$High_x(y) = \sum_{w=0}^{k-1} a_{x,w} y^w \bmod q$$

6. The proxy node x generates t shadows as

   $$S_{x,y} = High_x(y) \qquad (1 \le y \le t), \text{ where the proxy node } x \text{ has one shadow } S_{x,x}.$$

7. For all proxy nodes except itself, the proxy node x ($\ne v$) calculates as

   $$E_{x,v} = x \parallel S_{x,v},$$

   where $\parallel$ indicates the *concatenation* of data and v indicates an ID number of a proxy node other than the proxy node x.
8. The proxy node x distributes F_x to all proxy nodes except itself, and sends $E_{x,v}$ secretly to a proxy node v.
9. From all proxy nodes except itself, proxy node $x(\ne v)$ receives F_v, $E_{v,x}$, thereby obtaining $S_{v,x}$.
10. The proxy node x calculates its own shadow S_x and the corresponding public key as

$$S_x = \sum_{v=1}^{t} S_{v,x} \bmod q,$$

where a system secret S is recovered from k shadows, and $Y_x = g^{S}{}_x \bmod p$.

11. For $v = 1, 2, ..., t$, the proxy node x checks all F_v and $S_{v,x}$ using a verification formula in the paper [P91]. This protocol terminates if the result is "invalid".
12. The proxy node x uploads its own public key Y_x and F_x to a public bulletin board.

Generation of a Node's Shadow.

This procedure is similar to the setup phase of the RSS scheme by Anzai et al.

1. The proxy node x decides a parameter k_x satisfying

   $$0 \le d_x \le k_x - 2 < n_x.$$

2. The proxy node x divides its own shadow S_x into $n_x + k_x - 1$ shadows by a threshold k_x using a well-known Shamir scheme as

   a. The proxy node x puts $b_0 = S_x$.

   b. The proxy node x defines the following equation:

   $$Low_x(y_x) = \sum_{w_x=0}^{k_x-1} b_{wx}\, y_x{}^{w}{}_x \bmod q,$$

   where $b_1, b_2, ..., b_{kx-1}$ are random integers that satisfy the following conditions:

   $$0 \le b_{wx} \le q - 1 \text{ for all } 1 \le w_x \le k_x - 1 \text{ and } b_{kx-1} \neq 0.$$

   c. The proxy node x generates $n_x + k_x - 1$ shadows as

   $$S_{xi} = Low_x(i) \qquad (i \in \Delta_x).$$

3. For $i \in \Delta_x$, the proxy node x securely distributes the shadow S_{xi} to node(s) i. Nodes retain their own shadows as a secret key.
4. For $i \in \Delta_x$, the proxy node x calculates public key Y_i using the following equation:

   $$Y_{xi} = g^{S}{}_{xi} \bmod p.$$

Then, proxy node x uploads all public keys to the public bulletin board with the corresponding nodes' ID numbers. The remaining $k_x - 1$ public keys and the corresponding spare low ID numbers are uploaded to the public bulletin board as spare public keys. In addition, the proxy node x can generate information to calculate the public keys from parameters $b_0, b_1, ..., b_{kx-1}$ in a similar manner to that described in step 4 of "Generation of a Proxy node's Shadow". In this manner, the proxy node x may upload information instead of the public keys.

Table 3.2.2 shows an illustration of the initialization on the parameters ($t = 3$, $n_1 = 3$, and $n_2 = 2$). Note that a system secret S is recovered by k shadows (S_x), and S_x is recovered by k_x shadows (S_{xi}).

Table 3. 2. 2. Illustration of initialization.

	Low-stage subgroup 1			Low-stage subgroup 2		
High-stage subgroup	Proxy node 1 (Node 1) S_1			Proxy node 2 (Node 2) S_2		Spare high ID number 3 (S_3)
	Node 4 $S1_4$	Node 5 $S1_5$	Node 6 $S1_6$	Node 7 $S2_7$	Node 8 $S2_8$	

3. 2. 2. 3. Revocation Phase

3. 2. 2. 3. 1. Distribution by a Sender

First, a sender generates high-stage revocation data $RD(\Lambda, R)$ and low-stage revocation data $RD_x(\Lambda_x, R)$ in the following manner.

Generation of High-stage Revocation Data.

1. The sender determines a revoked proxy node(s). To revoke a proxy node x is equivalent to revoking the subgroup x. Here, d is the number of revoked proxy nodes.
2. The sender chooses R ($\in \boldsymbol{Z}_q$) at random and picks k - d - 1 integers from Γ, and let Θ be the set of chosen integers. Then, the sender calculates k - 1 the following high-stage revocation data:

   $$M_j = Y_j^R \bmod p \qquad (j \in \Lambda \cup \Theta),$$

   using the public keys corresponded with the revoked proxy nodes and the corresponding spare public keys on the public bulletin board. In addition, public keys that correspond to spare high ID numbers can be calculated from all F_x, using a method shown on the paper [KD98].
3. The sender calculates the following common data:

   $$X = g^R \bmod p.$$

Generation of Low-stage Revocation Data.

1. The sender determines a revoked node(s) from each low-stage subgroup. Here, d_x is the number of the revoked node(s).
2. The sender picks k_x - d_x - 1 integers from Γ_x, and lets Θ_x be the set of the chosen integers. Then, the sender calculates k_x - 1 the low-stage revocation data as follows:

   $$M_{xj} = Y_{xj}^R \bmod p \qquad (j \in \Lambda_x \cup \Theta_x),$$

   using the public keys corresponding with the revoked nodes and the corresponding spare public keys on the public bulletin board. In addition, he can calculate the public keys from information on the public bulletin board.
3. The sender constructs the revocation data $RD(\Lambda, R)$ and $RD_x(\Lambda_x, R)$ as

$$RD(\Lambda, R) = X \,\|\, \{[j, M_j] \mid j \in \Lambda \cup \Theta\},$$
$$RD_x(\Lambda_x, R) = X \,\|\, \{[j, M_{xj}] \mid j \in \Lambda_x \cup \Theta_x\}.$$

4. The sender distributes $RD(\Lambda, R)$ to the entire subgroup and distributes $RD_x(\Lambda_x, R)$ to the corresponding low-stage subgroup x.

Generation of Group Key.

Next, the sender calculates a group key U using all the proxy node public keys and secret random number R as

$$\begin{aligned} U &= \prod_{i \in \Phi} (Y_i^{L(\Phi, i)})^R \bmod p \\ &= g^{R \cdot \Sigma_{i \in \Phi} S_i \cdot L(\Phi, i)} \bmod p \\ &= g^{R \cdot S} \bmod p. \end{aligned} \qquad (3.2.1)$$

where

$$L(\Psi, W) = \prod_{T \in \Psi \setminus \{W\}} T / (T - W) \bmod q. \qquad (\forall \Psi: \text{set}, \forall W: \text{integer})$$

3. 2. 2. 3. 2. Receiving by a Proxy node

Receiving the revocation data $RD(\Lambda, R)$, a non-revoked proxy node x calculates a group key U using secret key S_x as follows:

$$U = X_x^{S_x \cdot L(\Lambda \cup \Theta \cup \{x\}, x)} \cdot \prod_{i \in \Lambda \cup \Theta} M_j^{L(\Lambda \cup \Theta \cup \{x\}, j)} \bmod p. \qquad (3.2.2)$$

The group key U is the same as the group key of the sender:

$$\begin{aligned} U &\equiv g^{R \cdot S_x \cdot L(\Lambda \cup \Theta \cup \{x\}, x)} \cdot \prod_{i \in \Lambda \cup \Theta} g^{R \cdot S_j \cdot L(\Lambda \cup \Theta \cup \{x\}, j)} && (\bmod p) \\ &\equiv g^{R\{S_x \cdot L(\Lambda \cup \Theta \cup \{x\}, x) + \Sigma_{j \in \Lambda \cup \Theta} S_j \cdot L(\Lambda \cup \Theta \cup \{x\}, j)\}} && (\bmod p) \\ &\equiv g^{R \cdot S} && (\bmod p). \end{aligned}$$

Every non-revoked proxy node obtains the same group key $g^{R \cdot S} \bmod p$, gathering k shadows on the exponent part in eq. (3.2.2).

On the other hand, a revoked proxy node j cannot calculate the group key U because M_j includes a secret key S_j of the revoked proxy node j, and the proxy node j can gather only k - 1 shadows.

3. 2. 2. 3. 3. Receiving by a Node

Receiving the revocation data $RD(\Lambda, R)$ and $RD_x(\Lambda_x, R)$, a non-revoked node v of a subgroup x calculates a group key U using personal secret key S_{xv} as follows:

1. The node v generates a high-stage data HD_x as

$$\begin{aligned} HD_x &= X_{xv}^{S_{xv} \cdot L(\Lambda_x \cup \Theta_x \cup \{v\}, v)} \cdot \prod_{i \in \Lambda_x \cup \Theta_x} M_{xj}^{L(\Lambda_x \cup \Theta_x \cup \{v\}, j)} \bmod p, \qquad (3.2.3) \\ &= g^{R \cdot S_x} \bmod p. \end{aligned}$$

2. The node v generates the group key U as

$$U = HD_x^{L(\Lambda \cup \Theta \cup \{x\}, x)} \cdot \prod_{i \in \Lambda \cup \Theta} M_j^{L(\Lambda \cup \Theta \cup \{x\}, j)} \bmod p. \qquad (3.2.4)$$
$$= g^{R \cdot S} \bmod p.$$

As a result, this group key U is the same as the group key of the sender and the proxy nodes.

3. 2. 2. 3. 4. Sending an Encrypted Message by a Sender

A sender can send an encrypted message using a group key U to all nodes except revoked nodes as follows:

the sender broadcasts the message to the group, and

the sender broadcasts the message to all proxy nodes (or each proxy node); subsequently, each proxy node broadcasts the message to its own subgroup (or sends the message to each node of its own subgroup).

3. 2. 3. Other Considerations

This section describes some considerations that are necessary to apply the proposed scheme to actual decentralized networks.

3. 2. 3. 1. How to Decide a Threshold *k*

The proposed scheme allows a sender to revoke a maximum of k - 2 proxy nodes and k_x - 2 nodes on each subgroup. In addition, the parameter k and k_x determines transmissions and calculation costs. Moreover, k and k_x are security parameters because a system secret key S is recovered. The number of nodes that the sender can revoke at once becomes large and the security becomes high if k and k_x are large; however the costs increase as they become large. Therefore, proxy nodes should determine the parameters k and k_x to fit actual networks.

3. 2. 3. 1. Node Join

A proxy node x decides a unique ID number c that satisfies $n_x + k_x \leq c \leq q - 1$ when a new node wants to join a low-stage subgroup x. The proxy node x calculates its secret key $S_{xc} = Low_x(c)$ and sends it to the new node securely. Then, the proxy node x calculates the corresponding public key y_c ($= g^{S}_{xc} \bmod p$) and adds it to a public bulletin board.

Using the above procedure, the corresponding proxy node can issue a new secret key for the node if that node loses his own secret key. However proxy nodes should re-initialize the system before the number of lost keys reaches a threshold k.

3. 2. 3. 2. Some Modifications

The basic scheme cannot revoke a single proxy node (i.e. the subgroup is also revoked). Therefore,

the study proposes a method to revoke a single proxy node. In an initialization phase, a node must belong to at least k - 1 subgroups because it is possible to revoke the node if the node belongs to less than k - 1 subgroups. In a revocation phase, a node of a revoked subgroup can calculate a group key from shadows of non-revoked subgroups. However, this modification increases the costs of key storage, transmission, and calculations concomitant with the number of subgroups to which the node belongs.

The proposed scheme consists of two stages (a high-stage and a low-stage). The proposed scheme is easily extendable to two more stages to share a node's shadow in the low-stage with nodes in a lower stage. In this case, the low-stage nodes inform the corresponding proxy node about a state (e.g. a node joins) of a lower stage.

The technique [AMM99-3] can modify equations (3.2.1)(3.2.2)(3.2.3)(3.2.4) to speed them up by reducing bits numbers of exponent parts in these equations: (3.2.1)(3.2.2)(3.2.3)(3.2.4). The study evaluates the technique experimentally to verify that the technique is most suitable for the proposed scheme.

3. 2. 4. Evaluation

3. 2. 4. 1. Security Analysis

First, this study discusses passive attacks, wherein a revoked node (or an outsider) uses only public data to get a group key U or secret parameters:

Getting R, S, S_x, or S_{xi}: The only secret parameters are R, S, S_x, and S_{xi} in exponent parts of data on a public bulletin board and a communication channel. Therefore, the difficulty of getting the secrets is equal to that of solving *Discrete Logarithm Problem*.

Getting U from g^R mod p and g^S mod p: All nodes can get X (= g^R mod p) and g^S mod p. Obtaining a group key U (= $g^{R \cdot S}$ mod p) from g^R mod p and g^S mod p is as difficult as solving *Diffie-Hellman Problem*.

Next, the study discusses active attacks:

Modifying or forging transmissions: All transmissions should be signed by a sender or a node to prevent modification and forgery. Moreover, this study considers that a time-stamp is necessary to prevent replay attacks for all transmissions.

Modifying or forging a public bulletin board: Modifying or forging a public bulletin board is difficult because this study assumes that the public bulletin board has a certain parameter with the corresponding certificates. The validity of the certificates shall be confirmed before using the parameters.

Publishing S_{xj} by a revoked node j: This study presumes that all revoked nodes publish their secret keys S_{xj}. Even if a non-revoked node uses her own secret key, she cannot calculate S_x

because she can only obtain $d_x + 1$ shadows that are less than a threshold k_x.

Conspiracy of nodes: This study assumes that a non-revoked node does not conspire with a revoked one. Even if all revoked nodes conspire, they cannot reconstruct a secret key Sx because they can only get, at most, d_x ($\leq k_x - 2$) shadows S_{xj}, which is less than the threshold value of k_x. Moreover, to get a system secret key S, the conspired nodes must get k secret keys of proxy nodes. On the other hand, this study assumes that proxy nodes are more trusted than nodes because the proxy nodes are elected by nodes. This study presumes that the proxy nodes do not publish their secret keys S_x and do not conspire.

Finally, the thesis discusses forward and backward confidentiality:

Forward confidentiality is that when a node is revoked, the node cannot obtain a group key of a next round,

Backward confidentiality is that when a new node joins a subgroup, the node cannot get a group key of a previous round.

Forward confidentiality is usually required in many systems. The proposed scheme satisfies forward confidentiality if it uses a method proposed in Section 4.3 of another paper [AMM01]. Backward confidentiality is not always required in actual systems. The proposed scheme satisfies backward confidentiality if it uses methods cited in previous papers [KMSW02-1][KMSW02-2].

3. 2. 4. 2. Performance

This section evaluates the features and performance of the proposed scheme by comparing it with four previously introduced schemes: **STW** [STW00], **NW** [NW04], **CK** [CK00], and **KMSW** [KMSW01]. In addition, this section evaluates the **basic scheme**, which is the proposed scheme, without adopting multistage secret sharing. In other words, the **basic scheme** is the RSS scheme by Anzai et al. initialized using the Pedersen scheme. Moreover, the section evaluates the technique [AMM99-3], then show that the technique is most suitable for the proposed scheme.

First, in Table 3.2.3, this study evaluates the features and the performance of the proposed scheme from the viewpoints of the following four requirements described before: **Decentralization**, **Independence**, **Efficiency**, and **Resistance**. Table 3.2.3 shows that only the proposed scheme satisfies the above four requirements, whereas four existing methods do not. Here, this study regards max revocation costs $O(n^2)$ as not satisfying efficiency.

Next, this section evaluates the technique. On nodes' operations of the technique and the RSS scheme by Anzai et al., the study increases threshold k one by one. Results of the examination show that costs of the technique are smaller than those of the RSS scheme in a certain range of k. The examination environment consists of CPU: Pentium3 1 GHz, Memory: 512 MB, and Compiler: VC++. Table 3.2.4 shows an available range and an operation time ratio of the technique to the RSS scheme for each parameter (| q | bit, | p | bit and n).

Table 3.2.3. Comparison of four requirements between the proposed scheme and other schemes.

	Decentralization	Independence	Efficiency			Resistance
			Key storage	Initialization costs	Max revocation costs	
Proposed scheme	Yes	Yes	1	$O(n_x^2)$	$O(n_x^2)$	Yes
Basic scheme	Yes	Yes	1	$O(n^2)$	$O(n)$	No
[STW]	No	No	1	$O(n)$	$O(n)$	Yes
[NW]	Yes	Yes	1	$O(n)$	$O(n^2)$	Yes
[CK]	Yes	No	$O(n_x)$	$O(n_x^2)$	$O(n_x^2)$	Yes
[KMSW]	No	Yes	$O(\log n)$	$O(n)$	$O(n)$	Yes

Table 3.2.4. Available range and operation time ratio of technique [AMM99-3] to scheme [AMM99-4].

	$n = 100$		$n = 1000$		$n = 10000$	
	$\|q\| = 160$	$\|q\| = 256$	$\|q\| = 160$	$\|q\| = 256$	$\|q\| = 160$	$\|q\| = 256$
	$\|p\| = 1024$	$\|p\| = 2048$	$\|p\| = 1024$	$\|p\| = 2048$	$\|p\| = 1024$	$\|p\| = 2048$
***k* range**	3 to 37	3 to 59	3 to 21	3 to 34	3 to 14	3 to 24
max *k*	9	11	7	9	5	7
max ratio	0.46	0.36	0.55	0.44	0.62	0.5

In Table 3.2.4, ***k* range** is the available range of k, **max ratio** is the largest ratio (i.e., in this case the technique is most efficient), and **max *k*** is a threshold in the case of **max ratio**. From Table 3.2.4, the technique is most efficient when $n = 100$. Therefore, the study infers that the technique is most suitable for the proposed scheme because the study assumes that the size of each subgroup is about $n_x = 100$.

3. 2. 4. 2. Ad-hoc networks

The proposed scheme is suitable for ad-hoc networks for the following reasons:

On the key exchange phase (high-stage), each proxy node must send the same key information to other proxy nodes.

On the key revocation phase, a sender and each proxy node must send same revocation data respectively to proxy nodes and nodes.

Using multi-hop communication, it is possible to reduce the transmission costs of the proposed scheme. In short, a sender or each proxy node need not copy the key information or the revocation data for every proxy node or every node.

3. 2. 5. Conclusion

This thesis proposed a user revocation scheme that is suitable for ad-hoc networks. The proposed scheme satisfies the following features: a privileged center is not required for initialization and user revocation; user revocation does not depend on communication routes; costs of revoking users and initialization are sufficiently small; and security against collusion is provided up to d_x revoked users.

3. 3. Interaction Key Generation Schemes

3. 3. 1. Introduction

3. 3. 1. 1. Background

Interaction key technology is based on a group key generation scheme and has the feature related to time-stamp technology. The group key generation scheme allows a group member to send a communication (is used to generate a shared key) over various channels to other group members, so that all the group members share the shared key. The channels could be Internet, LANs, cell phone networks, P2P networks, ad-hoc networks, mesh networks, etc. In these systems, there is a need for a secure and fast method to share shared keys with all other authorized group members. Also, there is a need for a time-stamp technology [HS90] using a trusted time-stamp authority can prove a message existed at a point in time. However, the interaction key technology does not combine the group key generation scheme and the time-stamp technology.

This paper focuses on a new feature whereby group members can prove to a third party that the members generate the shared key in real time, and the party can generate the group public key (which is called *Interaction key* in this paper) corresponds to the shared key. This third party (which is called *Proxy node* in this paper) can observe the communications of the group members. Normally, the proxy node cannot generate a trusted public key since he does not know the relationship between the communicating group members and the shared key. The simple solution is to assume that a trusted authority generates the shared key and produce a corresponding public key with a time-stamp and distribute them to the group members. However this solution is not practical.

Since the proposed solution does not require an authority, this study will discuss ordinary key generation schemes.

3. 3. 1. 2. Related Works

Diffie-Hellman key generation scheme and its many variations are well known, whereby two entities (e.g. Alice and Bob) share a shared key. M. Burmester and Y. Desmedt [BD94] extended this scheme to more than two entities. Their scheme does not restrict the use of the shared key, which is generally used for a symmetric cipher and MAC.

SSL [SSL] and TLS [TLS] provide a secure session between a proxy node and a node using a key generation scheme that distributes encrypted shared key information using public key cryptography. SSL and TLS use the shared key for a symmetric cipher and MAC.

In these schemes, in order to use the shared key as a private key for public key cryptography,

Alice and Bob can publish the public keys correspond to the shared key, (where the public keys are calculated by a procedure that Alice and Bob decide in advance) then a third party can verify to equal all the public keys. In this paper, this solution is called *Simple method*.

Two schemes [G99] [PS98] propose that Alice and Bob cooperate to generate identical RSA public keys and different private keys as a part of RSA private key. Another [BF97] shows how more than two entities cooperate to generate similar keys. In [G99] [PS98] Alice and Bob can generate the public keys using their secret keys. In [BF97] an RSA public key can be generated only if a Helper assists. However the Helper cannot access the private key. In other words, a third party cannot get the public keys. In order for a third party to get the public keys, it can use the *Simple method*.

Other related applications using a group key include Group Signature, Ring Signature, and Threshold Cryptosystem. Chaum and van Heyst [CH91] introduced the idea of *Group Signature* with the following features:

only group members can sign message, and

no one is able to identify who member of the group signed, however, in case of disputes, it is revealed the identity of the group member who signed it.

Unlike Group Signature, *Ring Signature* (Rivest, Shamir, and Tauman [RST01]), maintains anonymity even in case of disputes. *Threshold Cryptosystem* is that n members share a secret key of a group such that k members ($1 \leq k \leq n$) must cooperate in order to decrypt a given ciphertext. Two efficient threshold schemes [DF89][P91] are known, [DF89] requires an authority, while [P91] does not.

The proposed schemes also generate a group public key, while there are other features not seen in the previous examples. Additionally, this study assumes that group members share the same secret (i.e. shared key), however their applications assume that each member has a different secret. Thus, the study does not need to compare the performance of the proposed schemes and one of their applications.

In PKI model, a certificate authority can prove a validity of a public key to a verifier. At the same time, by observing key generation process, these proposed schemes can prove directly a validity of a prover's group public key to a verifier without an authority.

3. 3. 1. 3. Simple Methods

Here this study will discuss the simple method and its problems in detail. Alice and Bob have the private keys x_{Alice} and x_{Bob} respectively. A verifier has the corresponding public keys y_{Alice} and y_{Bob}. Also, Alice and Bob use the same procedure to calculate a public key from a shared key.

1. Alice and Bob share the shared key using a key generation scheme (e.g. Deffie-Hellman).
2. Alice and Bob calculate the public keys PK_{Alice} and PK_{Bob} from the shared key respectively.

3. Alice calculates the signature SIG_{Alice} for PK_{Alice} using X_{Alice}. Similarly Bob calculates the signature SIG_{Bob}.
4. Alice sends PK_{Alice} and SIG_{Alice} to Verifier. Similarly Bob sends PK_{Bob} and SIG_{Bob}.
5. A verifier verifies SIG_{Alice} and SIG_{Bob}. If the verification fails, the process is stopped.
6. If $PK_{Alice} = PK_{Bob}$ Verifier accepts PK_{Alice} and PK_{Bob} as the group public key.

The simple method has the following problems:

Problem 1: A verifier cannot know when the shared key is generated, and who shares it.

Problem 2: A verifier cannot believe PK_{Bob} (or PK_{Alice}) if Alice (or Bob) did not publish PK_{Alice} (or PK_{Bob}.

Problem 1 means entities other than Alice and Bob can share the shared key and the age of the key is completely unknown. Problem 2 means although Alice (or Bob) may have the shared key, it is possible that Alice (or Bob) is not included in the group.

3. 3. 1. 4. Our Result

The study hereby proposes solutions for both problems. The proposed schemes (TC type and DH type) feature:

Generatability is that a proxy node can generate an interaction key from communications (to generate a shared key).

Verifiability is that a proxy node can verify a relation between the interaction key and the communications, i.e., the information of the shared key (corresponds to the interaction key) is included in the communications.

Robustness is that a proxy node can certainly calculate the interaction key, when all users can calculate the shared key.

Confidentiality is that there is no polynomial time algorithm that can calculate the shared key from only the communications.

Generatability and Verifiability solve Problem1, Generatability and Robustness solve Problem2. Confidentiality assures that attackers cannot get the shared key.

For proper functionality of the proposed schemes, the study makes the following assumption:

At least one of users who participates in the proposed schemes is honest, also an honest user follows the proposed schemes.

If the number of users is large, this condition is generally satisfied. If not, this condition is satisfied, for instance, in applications such as E-commerce where users are on opposite sides of a transaction. In other words, since a seller and a buyer are on opposite of a transaction in general, the study considers that each side can use the proposed schemes to verify the other.

Since the proposed schemes can solve the problems of the simple method, the proposed schemes can be applied to many systems. There are many types of networks that function without an authority. Some of these are P2P networks, mesh networks, and ad-hoc networks. Unlike traditional key generation schemes and message authentication schemes, it is easy to apply the proposed schemes to these types of networks since the schemes prove a validity of a group public key without assuming an authority. Mesh and wireless ad-hoc type networks are especially suited to the proposed schemes, since an end user (i.e. proxy node) forwards packets. Note that a proxy node is a verifier, but is not an authority.

This study uses *threshold cryptosystems* [DF89][P91] based on *Discrete Logarithm Problem* which is called *DLP* in this paper) for our TC type proposed scheme. This is a type of secret sharing scheme wherein the user can verify the validity of the shadows. This means that Verifiability can be satisfied. Also, this study uses *Diffie-Hellman Problem* [DH76] (which is called *DHP* in this thesis) for our DH type proposed scheme. DHP is, for given g^a and g^b, to compute g^{ab}, where let G be a cyclic group with generator g. If the study uses DHP for verifying communications to generate a shared key, the study expects to ensure Verifiability. In both type proposed schemes, each user publishes $g^{a \ user's \ secret}$ and generates the shared key from all the secrets. This will ensure Generatability.

3. 3. 2. Definitions

The proposed schemes can be based on appropriate *DHP* and *DLP* defined over finite cyclic groups, including subgroups of Jacobians of elliptic curves and so on. This thesis explains the proposed schemes over a prime field $\boldsymbol{Z}_p$.

A target system consists of the following:

System manager: A trusted party (i.e. station) that decides system parameters and sets each user *i*'s private key x_i and the corresponding public key y_i. Also it manages a public bulletin board that keeps system parameters and public keys for all users. For simplicity, this study uses the same key pair for public key cryptosystems and signature schemes. However the key pair should be different in practice.

Node *i*: A user labeled *i* as its ID number is a member of the group. To generate a shared key, users communicate with other users. Assume the total number of users is n. Let $\Phi = \{1,2, ..., n\}$ be the set of the users. In DH type proposed scheme, users consist of **Starter**, **Relay**, and **Terminator**. In a ring type network, Starter is user 1 that begins the protocol, Terminator is user n that ends the protocol, and Relay is zero or more user(s) that exists between Starter and Terminator. If $n = 2$, this means there is no Relay.

Proxy node: One or more proxy nodes can use polynomial time algorithms α and β. α can

calculate an interaction key *IK* from communications to generate a shared key, and β can verify the relation between *IK* and the communications, i.e., a shared key (corresponds to *IK*) is included in the communications. Also a proxy node can observe communications between users at all times. Note that a proxy node is not necessary as is an authority.

Interaction key is a group public key generated using an algorithm α. Here, the interaction key is the group public key that corresponds to the shared key *CK* as private key, where all users who participates in our protocol, share the shared key.

Next, the study makes the following system assumptions:

1. All users trust the system manager. The system manager does not operate anything illegal.
2. In TC type proposed scheme, all users have simultaneous access to the broadcasted data. This is called a broadcast network in this thesis.
3. In DH type proposed scheme, user i can send data to next user $i + 1$, the last user n then send one to user 1. This is called a ring type network in this paper.
4. All users can access to the public bulletin board at any time.

Next, the study makes the following security assumptions:

1. *DHP* and *DLP* are computationally hard to solve.
2. In $(n + 1, 2n)$ threshold cryptosystems, anyone with less than $n + 1$ shadows cannot get any information about the secret *CK*.
3. At least one of users who participates in the proposed schemes is honest, also an honest user follows the proposed schemes.
4. The system manager manages the public bulletin board strictly reject any change. Or, each public bulletin board parameter with the certificate shall be checked before use. Otherwise, each user may send its own public key certificate to another user, and they may share public parameters before starting the proposed scheme.
5. The broadcast network and the ring type network are not secure, i.e., anyone can see the data following across the networks.

As mentioned in Section 3.3.1, the proposed schemes must feature:

1. *Generatability*: There is a polynomial time algorithm α.
2. *Verifiability*: There is a polynomial time algorithm β.
3. *Robustness*: When all users can calculate the shared key *CK*, a proxy node can certainly calculate the interaction key *IK*.
4. *Confidentiality*: There is no polynomial time algorithm that can calculate *CK* from only the communications to generate the shared key.

3. 3. 3. The Proposed Schemes

This proposed schemes contain two phases: system setup phase and key generation phase.

3. 3. 3. 1. System Setup Phase

At first, a system setup phase is carried out only once. Also, this phase is the same in both type proposed scheme.

1. The system manager decides the following system parameters and publishes them to the public bulletin board:

   p: a large prime number

   q: a large prime number such that $q \mid p - 1$, and $2n < q$ in TC type proposed scheme.

   g: a q^{th} root of unity over $\boldsymbol{Z}_p$

   A_Enc(y, m): a secure asymmetric encryption function, which outputs a ciphertext ct of the message m using a public key y. Here, ct is decrypted by using the corresponding private key x.

   S_Enc(ck, m): a secure symmetric encryption function, which outputs a ciphertext ct of the message m using a shared key ck. Here, ct is decrypted using the shared key ck.

   H(m):a secure one-way hash function, which outputs a hash value of the message m. Here, the size of the hash value is same as that of a shared key.

   A_Sign(x, m): a secure signature generation function, which outputs a signature SIG_m of the message m using a private key x. Here, the validity of the signature SIG_m is checked using a public key y.

2. The system manager generates user i's public key y_i and its private key x_i for 1, 2, ..., n.
3. The system manager distributes the user's key pairs to each user 1, ..., n respectively in a secure manner. Each user keeps its own key pair as its secret key.
4. The system manager publishes y_1, ..., y_n on the public bulletin board with the corresponding user's ID numbers.
5. The system manager may remove the private keys x_1, ..., x_n after the system setup phase. The system manager's other tasks are to maintain the public bulletin board and to generate a secret key and a public key for each new user.

3. 3. 3. 2. Key Generation Phase: TC Type Proposed Scheme

3. 3. 3. 2. 1. Procedure of User i

A user sends “Reject” to all entities if a signature is “invalid”, then the protocol stops.

Step 1

1. Node i pre-generates a secret random number R_i ($\in \boldsymbol{Z}_q$), and stores it secretly.

2. Node i pre-generates random integers $a_{i,1}, a_{i,2}, ..., a_{i,n}$ that satisfy the following conditions:

   $a_{i,1} = R_i$, $0 \le a_{i,k} \le q - 1$ for all $2 \le k \le n$, and $a_{i,n} \neq 0$.

3. Node i pre-calculates as follows:

   $$F_i = \{F_{i,k} (= g^{a}{}_{i,k} \bmod p) \mid \text{for } k = 1, 2, ..., n\}. \qquad (3.3.1)$$

4. Node i defines the following equation over $\boldsymbol{Z}_q$:

   $$f_i(w) = \sum_{k=1}^{n} a_{i,k} w^k \bmod q.$$

5. Node i generates $2n$ shadows using a well-known secret sharing scheme [S79] as follows:

   $CK_{i,j} = f_i(j)$ $\quad (1 \le j \le 2n)$,

   where user i has two shadows $CK_{i,2i-1}$, $CK_{i,2i}$.

6. Node i calculates as follows:

   $SIG_{i,all,F} = sign(x_i, i \parallel F_i)$, where $\parallel$ indicates *concatenation* of data.

7. For all users except oneself, user i ($\neq v$) calculates as follows:

   $ENC_{i,v} = \text{A\_Enc}(y_v, i \parallel CK_{i,2v-1} \parallel CK_{i,2v})$, and

   $SIG_{i,v,ENC} = \text{A\_Sign}(x_i, i \parallel ENC_{i,v})$.

8. Node i broadcasts F_i and $SIG_{i,all,F}$, and sends $ENC_{i,v}$ and $SIG_{i,v,ENC}$ to user v.

Step 2

1. From all users except oneself, user i ($\neq v$) receives F_v, $SIG_{v,all,F}$, $ENC_{v,i}$ and $SIG_{v,i,ENC}$.
2. Node i verifies $SIG_{v,all,F}$ and $SIG_{v,i,ENC}$ using y_v. If $SIG_{v,all,F}$ and $SIG_{v,i,ENC}$ are "valid", he decrypts $ENC_{v,i}$ using x_i so that he gets $CK_{v,2i-1}$ and $CK_{v,2i}$.
3. Node i calculates as follows:

   $$CK_{2i-1} = \sum_{v=1}^{n} CK_{v,2i-1}\} \bmod q,$$

   $$CK_{2i} = \sum_{v=1}^{n} CK_{v,2i} \bmod q, \text{ and}$$

   $$IK_i = g^{CK}{}_{2i-1} \bmod p.$$

4. For IK_i and CK_{2i}, user i calculates as follows:

   $SIG_{i,all,IK \parallel CK} = \text{A\_Sign}(x_i, i \parallel IK_i \parallel CK_{2i})$.

5. Node i broadcasts IK_i, CK_{2i}, and $SIG_{i,all,IK \parallel CK}$.

Step 3

1. From all users except oneself, user i ($\neq v$) receives IK_v, CK_{2v}, and $SIG_{v,all,IK \parallel CK}$.
2. Node i verifies $SIG_{v,all,IK \parallel CK}$ using y_v. If $SIG_{v,all,IK \parallel CK}$ is "valid", he accepts IK_v and CK_{2v}.
3. Let Ω be a set of IDs of $n + 1$ shadows that user i has, user i calculates as follows:

   $$CK = \sum_{h \in \Omega} (CK_h \cdot L(\Omega, h)) \bmod q,$$

where

$L(\Omega, h) = \prod_{t \in \Omega \setminus \{h\}} t / (t - h) \bmod q$, and

$CK \equiv \sum_{i=1}^{n} R_i \pmod{q}$.

3. 3. 3. 2. 2. Procedure of a Proxy node

1. For i = 1, 2, ..., n, the proxy node verifies $SIG_{i, all, F}$ and $SIG_{i, all, IK \| CK}$ using y_i. If $SIG_{i, all, F}$ and $SIG_{i, all, IK \| CK}$ are "valid", he accepts F_i, IK_i and CK_{2i}. If they are "invalid", he broadcasts "Reject" and IK is not acceptable.
2. The proxy node checks the following verification formula in [P91]:

   $g^{CK_j} \equiv \prod_{i=1}^{n} (\prod_{k=1}^{n} F_{i,k}{}^{j})^{k} \pmod{p}$. (3.3.2)

   If the above equation (3.3.2) is "invalid", he broadcasts "Reject" and IK is not acceptable.
3. Let $\Delta = \{1, 2, ..., 2n\}$ be a set of the shadow's ID, the proxy node calculates as follows:

   $IK = \prod_{i=1}^{n} (IK_i^{L(\Delta, 2i - 1)} \cdot g^{CK2i \cdot L(\Delta, 2i)}) \bmod p$,

   where

   $IK \equiv g^{\wedge}(\sum_{i=1}^{n} R_i) \pmod{p}$.

3. 3. 3. 3. Key Generation Phase: DH Type Proposed scheme

3. 3. 3. 3. 1. Procedure of User i

Let $\Lambda_i = \{2, ..., i\}$ be a set, and let $\eta_i = \{1, i+2, ..., n\}$ be a set. A user sends "Reject" to all entities if a signature is "invalid", then the protocol stops.

Round 1

1. Starter (user 1) pre-generates a random integer R_1, D_1 ($\in \boldsymbol{Z}_q$), and pre-calculates as follows:

   $IK_1 = g^{R}{}_1 \bmod p$ and

   $CHA_1 = g^{D}{}_1 \bmod p$.

   Next, for a part of the interaction key IK_1 and the challenge CHA_1, he calculates as follows:

   $SIG_{1, IK} = \text{A\_Sign}(x_1, 1 \| IK_1)$, and $SIG_{1, CHA} = \text{A\_Sign}(x_1, 1 \| CHA_1)$.

   Then he sends IK_1, CHA_1, and $SIG_{1, IK \| CHA}$ to Relay (user 2), in case of n = 2, to Terminator (user 2).
2. Relay (user i = 2, ... , n - 1) pre-generates a random integer R_i ($\in \boldsymbol{Z}_q$), and pre-calculates as follows:

   $IK_i = g^{R}{}_i \bmod p$.

Next, he verifies $SIG_{1,\,CHA}$ using y_1, and calculates as follows:

$RES_{\mathrm{i}} = CHA_1{}^{R}{}_{i} \bmod p$,

$SIG_{i,\,IK} = \mathrm{A\_Sign}(x_{\mathrm{i}},\ i \parallel IK_i)$, and

$SIG_{i,\,RES} = \mathrm{A\_Sign}(x_i,\ i \parallel RES_i)$.

Then he sends CHA_1, $SIG_{1,\,CHA}$, a part of the interaction key IK_{i}, the signature $SIG_{i,\,IK}\}$, the response RES_{i}, and the signature $SIG_{i,\,RES}$ to Relay (next user $i + 1$), in case of $n = i + 1$, to Terminator (user n).

3. Terminator (user n) pre-generates a random integer R_{n}, D_{n} ($\in \boldsymbol{Z}_q$), and pre-calculates as follows:

   $IK_{\mathrm{n}} = g^{R}{}_{n} \bmod p$, and

   $CHA_{\mathrm{n}} = g^{D}{}_{n} \bmod p$.

   Next, he verifies $SIG_{1,\,CHA}$ using y_1, and calculates as follows:

   $RES_n = CHA_1{}^{R}{}_{n} \bmod p$,

   $SIG_{n,\,IK} = \mathrm{A\_Sign}(x_n,\ n \parallel IK_n)$, and

   $SIG_{n,\,RES} = \mathrm{A\_Sign}(x_n,\ n \parallel RES_n)$.

   Then he sends the challenge CHA_n, the signature $SIG_{n,\,CHA}$, a part of the interaction key IK_n, the signature $SIG_{n,\,IK}$, the response RES_n, and the signature $SIG_{n,\,RES}$ to Starter (user 1).

Round 2

1. Starter (user 1) verifies $SIG_{n,\,CHA}$ using y_n, and calculates as follows:

   $RES_1 = CHA_n{}^{R}{}_{1} \bmod p$, and

   $SIG_{1,\,RES \parallel D} = \mathrm{A\_Sign}(x_1,\ 1 \parallel RES_1 \parallel D_1)$.

   Then he sends the response RES_1, the random integer D_1, and the signature $SIG_{1,\,RES \parallel D}$ to Relay (next user $i + 1$), in case of $n = 3$, to Terminator (user 3).

2. In case of $i = 2$, Relay (user 2) calculates as follows:

   $AS\_ENC_i = \mathrm{A\_Enc}(y_{i+1}),\ \mathrm{i} \parallel \sum_{i \in \Lambda_i} R_i \bmod q)$, and

   $SIG_{i,\,AS\_ENC} = \mathrm{A\_Sign}(x_i,\ i \parallel AS\_ENC_i)$.

   Then he sends the ciphertext AS_ENC_i and the signature $SIG_{i,\,AS\_ENC}$ to Relay (next user $i + 1$), in case of $n = i + 1$, to Terminator (user n). In case of $i \neq 2$, Relay (user i) verifies $SIG_{i\text{-}1,\,AS\_ENC}$ using $y_{i\text{-}1}$, and decrypts $AS\_ENC_{i\text{-}1}$ using x_i so that gets $\sum_{\Lambda i} R_i \bmod q$. Then he calculates AS_ENC_i, $SIG_{i,\,AS\_ENC}$ and sends them to Relay (next user $i + 1$), in case of $n = i + 1$, to Terminator (user n).

3. Terminator (user n) verifies $SIG_{n\text{-}1,\,AS\_ENC}$ using $y_{n\text{-}1}$, and decrypts $AS\_ENC_{n\text{-}1}$ using x_{n} so that gets $\sum_{i \in \Lambda n\text{-}1} R_i \bmod q$. Next he calculates as follows:

   $AS\_ENC_n = \mathrm{A\_Enc}(y_1,\ n \parallel \sum_{i \in \Phi \setminus \{1\}} R_i \bmod q)$, and

   $SIG_{n,\,AS\_ENC \mid D} = \mathrm{A\_Sign}(x_{\mathrm{n}},\ n \parallel AS\_ENC_{\mathrm{n}} \parallel D_{\mathrm{n}})$.

Then he sends the ciphertext AS_ENC_n, the random integer Dn and the signature $SIG_{n,\, AS\_ENC \,\|\, D}$ to Starter (user 1).

Round 3

1. Starter (user 1) verifies $SIG_{n,\, AS\_ENC \,\|\, D}$ using y_n, and decrypts AS_ENC_n using x_1 so that gets $\sum_{i \in \Phi \setminus \{1\}} R_i \bmod q$. Then he calculates as follows:

   $AS\_ENC_1 = \mathrm{A\_Enc}(y_2, 1 \,\|\, \sum_{i \in \Lambda_1 \setminus \{2\}} R_i \bmod q)$, and

   $SIG_{1,\, AS\_ENC} = \mathrm{A\_Sign}(x_1, 1 \,\|\, AS\_ENC_1)$.

   Next he sends the ciphertext AS_ENC_1 and the signature $SIG_{1,\, AS\_ENC}$ to Relay (user i), in case of $n = 2$, to Terminator (user 2). Finally, he calculates the shared key $CK = \sum_{i \in \Phi} R_i \bmod q$ and verifies $g^{CK} \equiv \Pi^n_{i=1} IK_i \pmod p$. If the above equation is not the congruence, he sends "Reject" to all users then CK is not accepted. If the equation is the congruence, he accepts CK.
2. In case of $i = 2$, Relay (user 2) verifies $SIG_{1,\, AS\_ENC}$ using y_1, and decrypts AS_ENC_1 using x_2 so that gets $\sum_{i \in \Lambda 1 \setminus \{2\}} R_i \bmod q$. Then he calculates as follows:

   $S\_ENC_i = \mathrm{S\_Enc}(\mathrm{H}(\sum_{i \in \Lambda_1} R_i \bmod q), 1 \,\|\, \sum_{i \in \eta_i} R_i \bmod q)$, and

   $SIG_{i,\, S\_ENC} = \mathrm{A\_Sign}(x_i, i \,\|\, S\_ENC_i)$.

   Next, he sends the ciphertext S_ENC_i and the signature $SIG_{i,\, S\_ENC}$ to Relay (next user $i + 1$), in case of $n = i + 1$, to Terminator (user n). In case of $i \neq 2$, Relay (user i) verifies $SIG_{i\text{-}1,\, S\_ENC}$ using $y_{i\text{-}1}$, and decrypts $S\_ENC_{i\text{-}1}$ using $\mathrm{H}(\sum_{i \in \Lambda i\text{-}1} R_i \bmod q)$ so that gets $\sum_{i \in \eta i\text{-}1} R_i \bmod q$. Then he calculates S_ENC_i, $SIG_{i,\, S\_ENC}$ and sends them to Relay (next user $i + 1$), in case of $n = i + 1$, to Terminator (user n). Finally, in case of $i = 2, \ldots, n - 1$, he calculates the shared key CK and verifies one, similarly to Starter.
3. Terminator (user n) verifies $SIG_{n\text{-}1,\, S\_ENC}$ using $y_{n\text{-}1}$, and decrypts $S\_ENC_{n\text{-}1}$ using $\mathrm{H}(\sum_{i \in \Lambda n\text{-}1} R_i \bmod q)$ so that gets $\sum_{i \in \eta n\text{-}1} R_i \bmod q$. Finally, he calculates the shared key CK and verifies one, similarly to Starter.

3. 3. 3. 3. 2. Procedure of a Proxy node

1. The proxy node verifies all signatures using all users' public keys. If even one of the signatures is "invalid", he sends "Reject" to all users and IK is not acceptable.
2. The proxy node checks the following published order:

   $RES_2, \ldots, RES_n, D_1, RES_1, D_n$

   If the order is invalid, he sends "Reject" to all users and IK is not acceptable.
3. The proxy node checks the following congruence expression:

   $$RES_1 \equiv IK_1{}^{D}{}_{n} \pmod p, \text{ and} \qquad (3.3.3)$$

   $$\prod_{i \in \Phi \setminus \{1\}} RES_i \equiv (\prod_{i \in \Phi \setminus \{1\}} IK_i)^{D}{}_{1} \pmod p. \qquad (3.3.4)$$

If the expression is not the congruence, he sends "Reject" to all users and *IK* is not acceptable.

4. The proxy node calculates the interaction key $IK = \prod_{i \in \Phi} IK_i \bmod p$.

3. 3. 4. Other Considerations

Next the study will describe some considerations necessary to apply the proposed schemes to an actual group communication.

How to decide the number of group members n

TC type proposed scheme can share a maximum of *n* users at one time. However *n* cannot be known in advance, thus the study needs to pre-compute the expected *m* ($\geq n$) pieces of $F_{i,k}$ shown in the equation (3.3.2).

How to decide a starter, relays, and a terminator

DH type proposed scheme needs to determine a starter, relays, and a terminator before execution. For preventing the conspiracy, it is best to select them at random.

3. 3. 5. Evaluation

3. 3. 5. 1. Validity

This section evaluates the features, performance, and security of proposed schemes. The proposed schemes fulfill the four requirements of Generatability, Verifiability, Robustness, and Confidentiality as follows:

1. The proposed schemes satisfy *Generatability* since $IK = \prod_{i \in \Phi} g^R{}_i \bmod p$ and $g^R{}_i \bmod p$ is published.
2. TC type proposed scheme satisfies *Verifiability*, since the proxy node can verify that the shadows are generated correctly using the equation (3.3.2) in [P91] and each user can obtain the shadows. DH type proposed scheme satisfies only a part of *Verifiability*, since a proxy node can verify user *i* has R_i and R_i is included in IK_i using the equation (3.3.3) (3.3.4). Note that given CHA_i ($= g^D{}_i \bmod p$) and IK_v ($= g^R{}_v \bmod$ p), to calculate RES_v ($= g^{D_i \cdot R}{}_i \bmod p$) is *DHP*. However he cannot verify user *i* has R_v of another user *v*.
3. The proposed schemes satisfy *Robustness*, since all IK_is have been published when all users can calculate the shared key.
4. The proposed schemes satisfy *Confidentiality* for the reasons mentioned in Section 3.3.5.2.

Table 3.3.1 shows that the worst-case performance of proposed schemes for each user. This study assumes that the amount of computation is the total number of exponentiations, a public key

encryption / decryption and a A_Sign / verification are one exponentiation each. Also the study assumes that the amount of transmission traffic of a signature is $|p|$, where $|p|$ means the size of p.

Table 3. 3. 1. Evaluation of the proposed two schemes.

	TC type proposed scheme	DH type proposed scheme
The amount of computation	$5n + 2$	11
The amount of transmission	$(3n + 1) \mid p \mid + \mid q \mid$	$9 \mid p \mid + \mid q \mid$
The number of transmission	$n + 1$	3
The type of security	DLP	DHP
The type of network	Broadcast	Ring

From Table 3.3.1, the study considers that the performance of DH type proposed scheme does not depend on the total number of group members n. The performance of TC type proposed scheme, however, is affected by n. Also, a proxy node requires $4n + n^3$ exponentiations for TC type proposed scheme, and $4n + 7$ exponentiations for DH type proposed scheme.

3. 3. 5. 2. Security Analysis

Passive attacks

First, the study will discuss passive attacks, wherein a proxy node or an outsider of the group uses only the public data to get CK or the secrets:

Determining *CK* from communications to generate a shared key: In TC type proposed scheme, a proxy node or an attacker can get IK_i (= $g^{CK}{}_i \bmod p$), IK (= $g^{CK} \bmod p$), and n shadows (i.e. the number of CK_i is n) from the communications. CK can be recovered using $n + 1$ shadows, since $n + 1$ is the threshold value of the secret sharing scheme. However, since the difficulty of getting CK_i from IK_i is the same as that of solving DLP, they cannot get $n + 1$ shadows. For the same reason, they also cannot determine CK from IK.

In DH type proposed scheme, a proxy node or an attacker can get IK_i (= $g^{R}{}_i \bmod p$), IK (= $g^{CK} \bmod p$), and n encrypted R_is from the communications. CK can be recovered using n R_is, since $CK = \sum_{i \in \Phi} R_i \bmod q$. However, since the encryption functions A_Enc(y, m) and S_Enc(y, m) are secure and the difficulty of getting R_i from IK_i is the same as that of solving DLP, they cannot get n R_is. For the same reason, they also cannot determine CK from IK.

Active attacks

Next, the study discusses the active attacks:

Modifying or forging communications: In the proposed schemes, all communications consist of the signatures generated using a secure signature function A_Sign(*s*, *m*) and the generator's ID number *i*. If an attacker modifies the signatures or *i*, the signature verification function outputs "invalid". Therefore, it is difficult to modify or forge the communications. Also, in DH type proposed scheme, to use the signatures prevents '*Man in the middle attack*' in ring type networks.

Modifying or forging a public bulletin board: Modifying or forging the public bulletin board is difficult because the study assumes that the system manager ensures that there are no changes. Or, each parameter on the board has a certificate, which is verified before using the public parameters.

Conspiracy: The study assumes that at least one of users who participates in the proposed schemes is honest, deceiving the proxy node is possible only if all users are dishonest and they conspire at the same time. If this happens, the proxy node accepts the public key generated in advance as an interaction key.

In TC type proposed scheme, deceiving the proxy node is difficult if only one honest user exists, since the honest user *i*'s $F_{i,1}$ (= $g^{R}{}_{i}$ mod p) cannot be modified by the signature, and the dishonest users cannot get R_i until the honest user *i* publishes his shadow (i.e. the dishonest users cannot select their random secrets depending on R_i).

In DH type proposed scheme, deceiving the proxy node is possible if a dishonest starter and a dishonest terminator conspire. They can generate their own random secrets using the honest Relay *i*'s R_{i}. Thus, DH type proposed scheme requires that at least the starter or the terminator be honest.

To use a key pair for signing and verifying: Application of proposed schemes means that (*IK*, *CK*) are used for verifying and singing. This study assumes that a general signature algorithm is used, provided it substitutes (*IK*, *CK*) for a key pair of the algorithm. The difference between (*IK*, *CK*) and the key pair is that (*IK*, *CK*) have a construction as the sum of random numbers. Since a signature algorithm (where the generated signature is divided as it can verify) may exist, the study recommends not using such a signature algorithm for the proposed schemes.

3. 3. 6. Applications

This study will examine two applications for proposed schemes, message certification and certificate generation.

Message Certification:

The study assumes that *Message Certification* means that a third party can verify the communication messages between users using a signature generated by an authority. Using the proposed schemes, users can prove the following to a third party (i.e. proxy node) without an authority:

signed messages are communicated just now, and
signed messages have not been communicated before.

To demonstrate, first the users generate an interaction key and a shared key. Next the users generate the signature using the shared key (and their own private key), and send communications containing this signature. Finally the proxy node verifies the signature using the interaction key (and the user's public key).

Certificate Generation:

The study assumes that *Certificate Generation* means that a certificate authority can issue a certificate, which guarantees the time when users generated the key pair. Unless the certificate authority generates a key pair, the above certificate cannot be issued without using the proposed schemes.

3. 3. 7. Conclusions

This thesis has proposed a new concept known as *Interaction key*, realized by two schemes (TC type and DH type). An interaction key is a group public key that corresponds to a shared key shared by multiple users. It has unique features that a proxy node can verify:

1. the shared key has been generated now, and
2. the shared key has not existed before.

Thus, multiple users can prove them to the proxy node.

3. 4. Incentive and PKI-supporting Mechanism

3. 4. 1. Introduction

3. 4. 1. 1. Wireless Multi-hop Communications

Wireless multi-hop communication is that one or more relay nodes relay messages from a sending node to a receiving node as bucket brigade. Mobile ad-hoc networks and multi-hop cellular networks exist as networks using wireless multi-hop communication.

A mobile ad-hoc network (see Figure 3.4.1) is composed of a set of autonomous nodes, and the nodes communicate with each other without backbone infrastructures. Mobile ad-hoc networks have the following advantages: 1) no requiring backbone infrastructures, 2) effective utilization of wave frequency bands using short-range communication and traffic distribution, and 3) easy extending of network area.

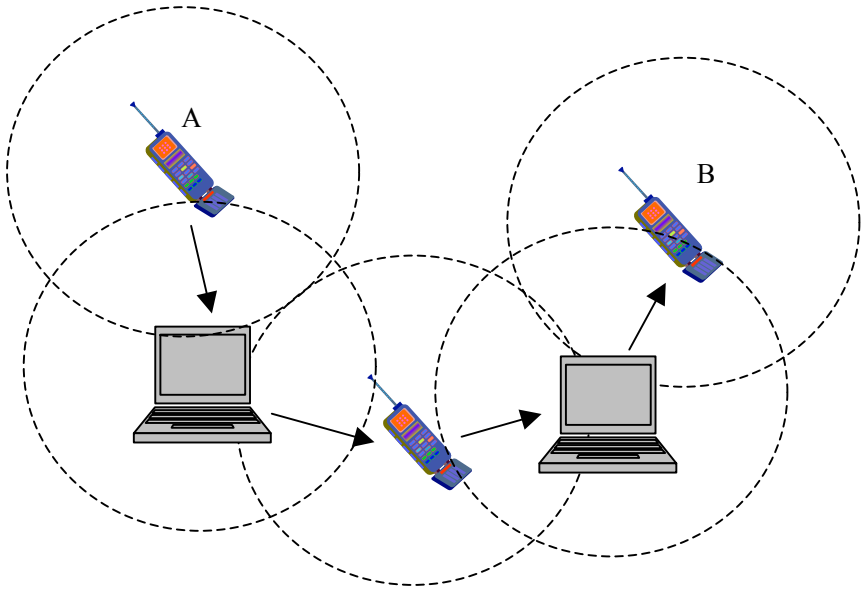


Fig. 3.4.1. Node A sends a message to node B using multi-hop communication in an ad-hoc network.

On the other hand, a multi-hop cellular network (see Figure 3.4.2) is an integrated network of a cellular network (e.g. a wireless LAN that connects to backbone infrastructures via access points, and cellular phone systems that connect to backbone infrastructures via base stations) and a mobile ad-hoc network. Here, nodes (i.e. cellular phones, notebook PC's, and PDAs) communicate with

backbone infrastructures using multi-hop communication. Therefore, this study would say a multi-hop cellular network is a cellular network, which adopts the advantages of ad-hoc networks. In this thesis, multi-hop cellular networks suppose as follows:

Wireless LAN systems and cellular phone systems

It is not always possible for a node to connect to stations (e.g. base stations and access points).

Node to node communication does not always have to go through a station.

At least one of node to station communications is multi-hop when node-to-node communication going through a station.

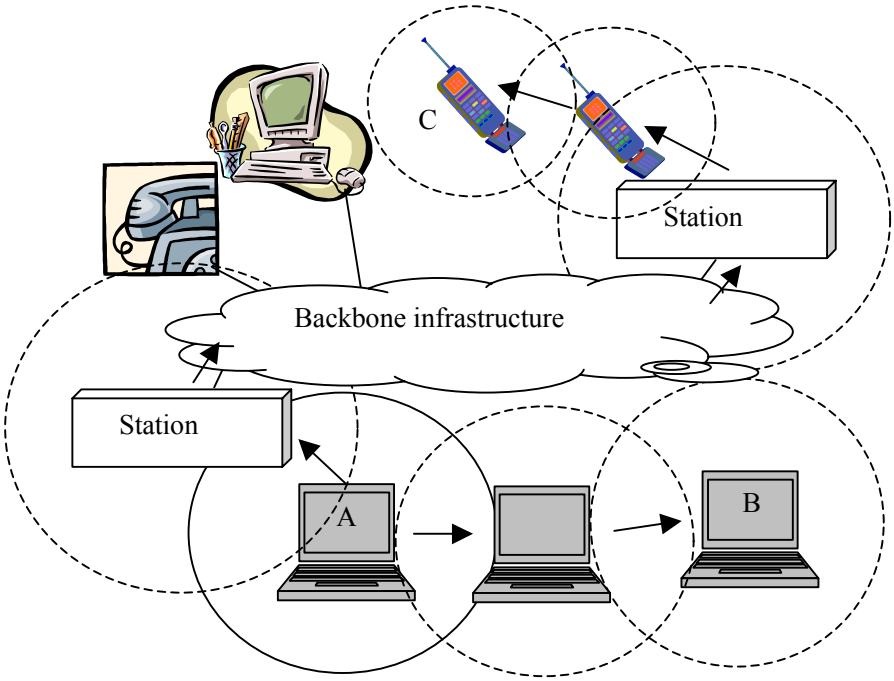


Fig. 3.4.2. Node A sends a message to node B and C in a multi-hop cellular network.

3. 4. 1. 2. Security of Multi-hop Communication

The thesis focuses on node-to-node authentication and cooperation of well-known security problems. As other problem, secure routing problems [BB02][MGLB00] are known.

3. 4. 1. 2. 1. PKI-supporting function

For the general public, entity authentication requires public key infrastructure (PKI). However PKI is not always effective by the following reasons:

A node has a heavy workload for certificate verifications and signature generations, since the node has not many resources. In addition, it is not easy that a node has various certification authority (CA) certificates for authentication.

A node does not always use CA for certificate management (but a close network needs not CA).

To solve the above problems, the study considers a PKI-supporting function that nodes with many resources manage certificates (certificate issuing, certificate providing, and certificate verification) for nodes with few resources.

3. 4. 1. 2. 2. Incentive Function

Multi-hop communication requires relaying messages to nodes. However, nodes are not active in cooperation with using power because of battery drive. Therefore, the study considers an incentive function that gives reward for cooperation and gives disadvantage for non-cooperation. Here, the thesis calls a digitizing reward a "reward point (*RP*)". Cooperation methods, which trade *RP* with services, have been studied extensively.

3. 4. 1. 3. Related Works

The thesis introduces studies that apply PKI to ad-hoc networks. The papers [CBH02][HBC01] proposed a modified PKI that trusts associates as with pretty good privacy (PGP). The modified PKI differs from PGP in that 1) nodes do not use public key directories, 2) nodes manage own certificate, and 3) nodes exchange certificates with other nodes. This proposal has a problem that nodes have a heavy workload.

On the other hand, the papers [KZLLZ01][YK02][ZH99] proposed schemes trust a group of privileged nodes. On the schemes, the group shares a CA's private key, and the group can manage certificates when the number of the shared keys is more than a specific threshold. The schemes have an issue in which a node must connect plural privileged nodes for certificate management. Also, the paper [WT01] proposed a scheme that trusts a group of non-privileged nodes. A node asks the group about trustworthiness of certificates by distributing the questions. Therefore, the scheme increases an amount of traffic. Moreover, a method to judge answers from the group is not clear in the paper.

Secondly, the thesis introduces studies of incentive functions. The paper [BH00][BH02] proposed a scheme that manages *RP* in a tamper resistant module (TRM) in ad-hoc networks. The scheme increases *RP* in TRM when relaying a message, and decreases *RP* in TRM when sending a message. Then a destination TRM sends *RP* to a source TRM if the relaying is valid. This scheme requires a high-performance TRM. Moreover, a method to send *RP* is not clear in the paper.

On the other hand, the papers [JHB03][SBHJ03] proposed a symmetric key cipher-based incentive scheme in multi-hop cellular networks. This scheme manages each node's *RP* account in a base station. After session establishing with the station, a sending node sends a message with a message authentication code (MAC) to a receiving node via the station. The base station removes *RP* of the sending node to the receiving node by verifying the message with MAC. In addition, the receiving node sends an advice of receipt to the station when the receiving node receives the

message. Nodes have a light workload. However the routing is not flexible, because the communication must go through the station and the scheme requires source routing method that a message includes IDs of relay nodes.

In addition, the paper [ZYC02] proposed an asymmetric key cipher-based incentive scheme in ad-hoc networks. A sending node sends a message with the corresponding signature to a receiving node via a relay node, and the receiving node and the relay node verify the received signature. The receiving node sends the signature to trusted third party (TTP), and then TTP removes *RP* of the sending node to the relay node. Therefore, the scheme requires TTP, also nodes have a heavy workload for verifying.

The papers [BB02][MGLB00] proposed secure routing schemes using node observation. On the schemes nodes observe messages relaying so that the nodes find a dishonest node, which does not relay a message. Then the schemes re-route without the dishonest node. Therefore, this study thinks to be able to use the schemes as an incentive function. For example, the scheme removes *RP* of a sending node to an honest relay node. However the message observation is a heavy workload for nodes.

3. 4. 1. 4. Overview

Existing papers discuss a PKI-supporting function and an incentive function respectively, therefore these papers do not investigate an integration of two functions. However an incentive scheme using PKI exists, conversely a PKI-supporting scheme using nodes cooperation exists. In a word, there are many cases where the functions are complementary to each other. It is important to discuss simultaneously about the two functions. This thesis proposes a mechanism by which a temporal privileged node (i.e. proxy node) provides PKI-supporting and incentive functions in multi-hop cellular networks. Specially, routing of this proposed scheme is more flexible than one of existing schemes [JHB03][SBHJ03].

3. 4. 2. System Structure and Design Policy

3. 4. 2. 1. System Structure

The proposed mechanism composes of the following entities:

Node: is a mobile terminal with wireless multi-hop communication technology (e.g. Bluetooth, ultra wide band (UWB), and IEEE802.11). A node has any node certificates (and the corresponding private key), any CA certificates, and TRM. This thesis supposes an IC card (e.g. subscriber identity module (SIM)) as TRM, and describes the IC card as a "*Card*". This proposed mechanism assumes mainly a cellular phone and PDA as a node.

Proxy node: is a temporal privileged node. A proxy node has a proxy certificate issued by a

system manager (and the corresponding private key), any node certificate (and the corresponding private key), a CA certificate that can verify a least one of plural system manager certificates, and TRM. Here, the proxy certificate has the short term of validity (e.g. one day), and indicates that the system manager approves the node as proxy node. The proposed mechanism assumes mainly a notebook PC as a proxy node, since a proxy should have more resources than a general node.

System manager: is a manager for a multi-hop cellular network, and .the thesis describes the manager as "*M*". This proposed mechanism assumes an organization, which provides a wireless LAN service or cellular phone service as a system manager. The system manager has almost CA certificates and his own certificates that plural certificate authorities issue. In addition all nodes trust the manager.

Station: is a base station of cellular phone service or an access point of a wireless LAN service.

Card issuer: is an issuer that issues a card (and the corresponding card application) and manages the card. The thesis describes the issuer as "*C*". Only the card issuer can increase *RP* in *Card*, and all entities trust the card issuer. In this thesis, a backbone infrastructure composes of a system manager and a card issuer.

3. 4. 2. 2. Design Policy

The proposed mechanism is that a proxy node (instead of a trusted center [JHB03][SBHJ03] [ZYC02]) provides PKI-supporting and incentive functions simultaneously. Our design policy satisfies as follows:

1. A message does not always go through a backbone infrastructure. Therefore the routing is flexible.
2. According to integrate of supporting and incentive functions, the proposed scheme can delete duplication processes (initialization, invalid node finding, and system protecting), and can improve security because of lumping invalid nodes and the corresponding information for two functions together.

For realizing the above policy 1, a backbone infrastructure must execute verifications for incentive function in non-real time. Therefore, the thesis presumes that a proxy node uses PKI.

Moreover our design policy satisfies the following properties:

Connectability: means a node can execute a certification verification process in strong probability.

Efficiency: means this proposed mechanism gives a little overhead to a system.

Verifiability: means a system can find an illegal operation.

Robustness: means invalid nodes collusion cannot leak a system secret.

Traceability: means a system can specify an invalid node.

3. 4. 3. The Proposed Mechanism

This section explains a preparation phase, a PKI-supporting function and an incentive function of this proposed mechanism. Here, the preparation phase is a common process that the two functions use for preparing. The proposed mechanism uses the following notation for the explanation:

H(*m*): is a secure one-way hash function, which outputs a hash value of the message *m*.

HMAC(*m*): is a secure HMAC function, which outputs a hash value of the message *m*.

ID_A: is an identification of entity A.

θ(*a, b, c*): is a reward point. "*a*" is ID. "*b*" is a point number. "*c*" is I/O type (In or Out). θ is input from *Card* when I/O type is "In", and θ is output to *Card* when I/O type is "Out". Note that θ exists in *Card* when I/O type is not described.

$AInfo_{A\ B}$: is authentication information that entity A sends to entity B. The authentication information is output of a HMAC function, witch is inputted a symmetric key $CKey_{A\ B}$ that entity A and B share and a target message. Entity A selects a 16 bytes random number and then he includes the random number the target message when he generates the authentication information. Entity B stops a protocol if a result that she verifies the authentication information is invalid.

3. 4. 3.1. Preparation Phase

In this proposed mechanism, a system manager approves a node as a proxy node *j*, and other nodes register themselves the proxy node *j* (see Figure 3.4.3). Then the proxy node *j* provides PKI-supporting and incentive functions to the registered nodes.

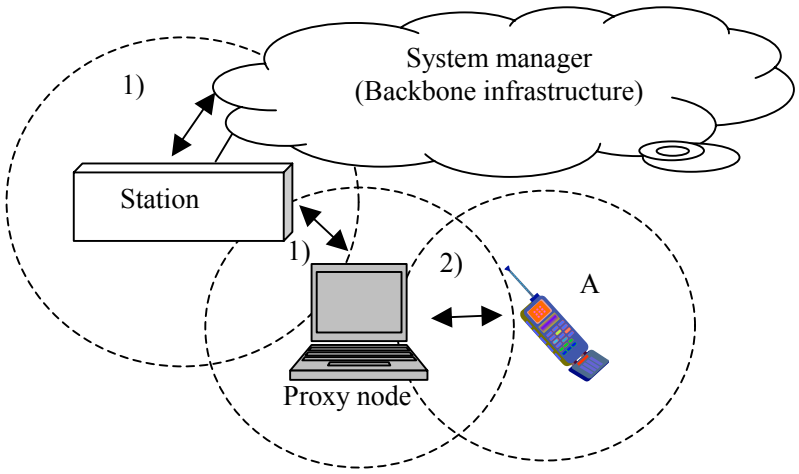


Fig. 3.4.3. 1) Approving a proxy node by a system manager, 2) Registering a node A to a proxy node

3. 4. 3. 1. 1. Approving a Proxy node

1. A node *i* sends his resource information $RInfo_i$ and the corresponding signature Sig_i generated by a private key $SKey_i$ (corresponds with his own certificate $Cert_i$) to *M*. Here, $RInfo_i$ includes

specifications (e.g. CPU, memory size, and HD size) by which the node *i* desires to publish, a subject list of CA certificates that he has, and $Cert_i$.

2. *M* issues a proxy certificate $PCert_i$ and transfer information $TInfo_i$ from $RInfo_i$ if a result that he verifies Sig_i is valid, and generates a symmetric key $CKey_{M\,i}$. Then *M* sends $Enc1_M$ and $Enc2_M$ to the node *i*. Here, $Enc1_M$ is a ciphertext by which *M* encrypts $PCert_i$, the corresponding private key $PSKey_i$, $TInfo_i$, and Sig_M using $CKey_{M\,i}$. Also $Enc2_M$ is a ciphertext by which *M* encrypts $CKey_{M\,i}$ using a public key of $Cert_i$. Sig_M is a signature that *M* signs $PCert_i$, $PSKey_i$, and $TInfo_i$ using a private key $SKey_M$ (corresponds with his own certificate $Cert_M$). $TInfo_i$ is PKI information that includes certificates of *M*, CA certificates, a certificate revocation list (CRL), and a black list of invalid nodes.
3. The node *i* decrypts $Enc2_M$ and $Enc1_M$ using a private key $SKey_i$ (corresponds with $Cert_i$) and $CKey_{M\,i}$, respectively. Then the node *i* accepts the decrypted messages if a result that he verifies Sig_M is valid.

3. 4. 3. 1. 2. Registering with a Proxy node

Registration is that a node *k* and a proxy node *j* mutual authenticate and then they share a symmetric key if a result of the mutual authentication is valid.

1. The node *k* sends $RInfo_k$, the corresponding signature Sig_k generated by using $SKey_k$ (corresponds with $Cert_k$) to the proxy node *j*. Note that the node *k* uses $TCert_k$ instead of $Cert_k$ if he has obtained a temporary node certificate $TCert_k$ (see step 2). Here, $TCert_k$ proves that the proxy node *j* has registered the node *k*.
2. The proxy node *j* generates $TCert_k$, (the corresponding private key $TSKey_k$,) and $CKey_{j\,k}$ if a result that he verifies Sig_k is valid. Next the proxy node *j* generates Sig_j using a private key $PSKey_j$ (corresponds with $PCert_j$ selected from $RInfo_k$) for the generated data. The proxy node *j* sends Sig_j, $Enc1_j$ (is a ciphertext that $TCert_k$ *and* $TSKey_k$ are encrypted by using $CKey_{j\,k}$), $Enc2_j$ (is a ciphertext that $CKey_{j\,k}$ is encrypted by using a public key of $Cert_k$), and $PCert_j$ to the node *k*.
3. The node *k* decrypts $Enc2_j$ and $Enc1_j$ using $SKey_k$ and $CKey_{j\,k}$, respectively. Then the node *k* accepts the decrypted messages if a result that he verifies Sig_j is valid. Here, the node *k* can request other CA certificates to a mutual authenticated node if the node *k* has not verifiable CA certificates.

There is every possibility of verifying a temporary node certificate than other certificates since an upper CA of the temporary node certificate is *M*.

The proxy node *j* can obtain *RP* from the system manager for providing services to the registered nodes. The proxy node *j* receives authentication information from the nodes when providing services. If the number of authentication information reaches to a threshold, the proxy node *j* sends the authentication information and the corresponding symmetric keys to *M*. Then *M* sends *RP* (is

issued by *C*) to the proxy node *j* if results that *M* verifies the authentication information are valid.

If the system assumes proxy node to proxy node communication, *M* gives a symmetric key (for sharing between proxy nodes) and other proxy nodes' information to proxy nodes when approving proxy nodes. For using the key and the information, a probability of node registration improves. In addition, it is possible to mutual authentication and key-sharing between registered nodes with proxy nodes.

3. 4. 3. 2. A PKI-supporting Function

This proposed mechanism supports proxy verification, certificate issuing, and public key directory service (i.e. certificates providing) as PKI-supporting function. This section explains the proxy verification and certificate issuing using mutual authentication and key sharing protocols.

3. 4. 3. 2. 1. Supporting Mutual Authentication and Key Sharing

The proposed mechanism realizes mutual authentication and key sharing (see Figure 3.4.4) between registered node A and B via a proxy node *j* using symmetric cryptography.

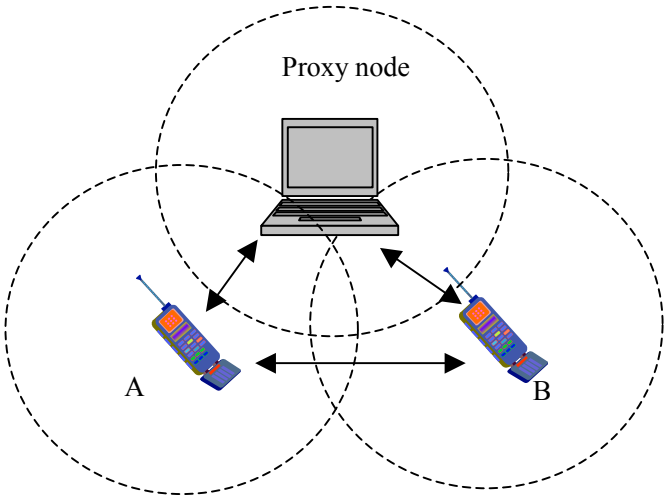


Fig. 3.4.4. Mutual authenticating and key sharing between a node A and B via a proxy node

1. The node A generates a random number $Rand_A$, and then he sends $ID_A \parallel Rand_A$ as a request message to the node B. Here, "||" is data concatenation.
2. The node B generates a random number $Rand_B$ if he responds to the request message, and then he sends $ID_B \parallel Rand_B$ to the node A.
3. The node A sends $Rand_A$, ID_B, the corresponding $AInfo_{A\,j}$, ID_A, and ID_j to the proxy node *j*.
4. The node B sends $Rand_B$, ID_A, the corresponding $AInfo_{B\,j}$, ID_B, and ID_j to the proxy node *j*.
5. The proxy node *j* sends $AInfo_{j\,A}$ (for $Rand_A \parallel Rand_B$), $Enc_{j\,A}$ (is a ciphertext that a symmetric key

$CKey_{A\,B}$ is encrypted using $CKey_{j\,A}$), ID_j, and ID_A to the node A if a result that he verifies $AInfo_{A\,j}$ using $CKey_{j\,A}$ is valid. In addition, the proxy node *j* sends $AInfo_{j\,B}$ (for $Rand_A \parallel Rand_B$), $Enc_{j\,B}$ (is a ciphertext that $CKey_{A\,B}$ is encrypted using $CKey_{j\,A}$), ID_j, and ID_B to the node A if a result that he verifies $AInfo_{B\,j}$ using $CKey_{j\,B}$ is valid.

6. The node A and B decrypt $Enc_{j\,A}$ and $Enc_{j\,B}$ using $CKey_{j\,A}$ and $CKey_{j\,B}$ respectively. Consequently, the nodes obtain $CKey_{A\,B}$ if results that the nodes verify $AInfo_{j\,A}$ and $AInfo_{j\,B}$ respectively are valid.

3. 4. 3. 2. 2. A Public Key Directory Service

A public key directory service is a service by which the proxy node *j* provides PKI information (e.g. certificates) to the registered nodes.

1. The node *i* sends a request list (e.g. a subject list of requested certificates), the corresponding $AInfo_{i\,j}$, ID_i, and ID_j to the proxy node *j*.
2. The proxy node *j* sends PKI information $PKIInfo_j$ (corresponds to the request list), the corresponding $AInfo_{j\,i}$, ID_j, and ID_i to the node *i* if a result that he verifies $AInfo_{i\,j}$ is valid.
3. The node *i* accepts $PKIInfo_j$ if a result that he verifies $AInfo_{j\,i}$ is valid. The node *i* should report a fact to *M* if contents of $PKIInfo_j$ are not equal to contents that are described in the request list, when the node *j* has provided *RP*.

3. 4. 3. 3. An Incentive Function

This subsection explains a *RP* management of *Card* and incentive models on an incentive function. For simplifying description, this thesis treats only proxy node *j*. Naturally, it allows to differ in a proxy node that is registered nodes and a proxy node that mediates *RP*.

3. 4. 3. 3. 1. RP Management in Card

The node *i* has $Card_i$ and can withdraw $\theta(i, X, \text{Out})$ of X ($\leq Y$) points from $\theta(i, Y)$ of $Card_i$. Here, $Card_i$ reduces $\theta(i, Y)$ by X points (i.e. $\theta(i, Y - X)$). On the other hand, only *C* can generate $\theta(i, L, \text{In})$ that increases $\theta(i, Y)$ of $Card_i$ by L points. This study assumes that $Card_i$ and *C* share a symmetric key $CKey_{i\,C}$ and they use $CKey_{i\,C}$ to generate / verify $\theta(i, X, \text{Out})$ and $\theta(i, L, \text{In})$. $Card_i$ rejects $\theta(i, L, \text{In})$ if $\theta(i, L, \text{In})$ is not generated by using $CKey_{i\,C}$. In addition, *C* should verify duplicate use as electric money technologies for preventing re-use of *RP*.

3. 4. 3. 3. 2. A Mutual Communication Incentive Model

A mutual communication incentive model is a model in which a providing node gives services to a requesting node *R* on a public key directory service. For instance, the node *R* requests certificates to the node *P*, and then the node *P* provides the requested certificates to the node *R*. Finally, the node *R*

gives *RP* to the node *P* as value. The study assumes that the node *R* and *P* mutual authenticate and they share a symmetric key $CKey_{P\,R}$.

1. The node *R* sends a request list, the corresponding $AInfo_{R\,P}$, ID_R, and ID_P to the node *P*. Here, the request list includes subjects of certificates if the node *R* desires to request certificates, and the list includes names if the node *R* desires to request a black list.
2. The node *P* sends an offer list, the corresponding $AInfo_{P\,R}$, ID_P, and ID_R to the node *R* if a result that he verifies $AInfo_{R\,P}$ is valid. Here, the offer list includes subjects of certificates or names black lists that the node.
3. The node *R* withdraws $\theta(R, T, \text{Out})$ of *T* points (corresponds the offer list) from $Card_R$ if a result that he verifies $AInfo_{P\,R}$ is valid and he is satisfied with the offer list. Then the node *R* sends $\theta(R, T, \text{Out})$, the corresponding $AInfo_{R\,P}$, ID_R, and ID_P to the node *P*. Note that nodes know points for contents (e.g. one certificate is 1 point) from *M* via proxy nodes in advance.
4. The node *P* sends contents (corresponds the offer list), the corresponding $AInfo_{P\,R}$, ID_P, and ID_R to the node *R* if a result that he verifies $AInfo_{R\,P}$ is valid. Then node *P* sends $\theta(R, \text{T}, \text{Out})$, the corresponding $AInfo_{P\,j}$, ID_P, and ID_j to the proxy node *j*.
5. The node *R* accepts the contents if a result that he verifies $AInfo_{P\,R}$ is valid.
6. The proxy node *j* sends $\theta(R, T, \text{Out})$, the corresponding $AInfo_{j\,M}$, ID_j, and ID_M to *M*. if a result that he verifies $AInfo_{P\,j}$ is valid.
7. *M* sends $\theta(R, T, \text{Out})$ to *C* if a result that he verifies $AInfo_{j\,M}$ is valid. Next *M* receives $\theta(P, T, \text{In})$ from *C*, and then he sends $\theta(P, T, \text{In})$, the corresponding $AInfo_{M\,j}$, ID_M, and ID_j to the proxy node *j*. Here, *C* reports a fact to *M* when $\theta(R, T, \text{Out})$ is invalid.
8. The proxy node *j* sends $\theta(P, T, \text{In})$, the corresponding $AInfo_{j\,P}$, ID_j, and ID_P to the node *P* if a result that he verifies $AInfo_{M\,j}$ is valid.
9. The node *P* inputs $\theta(P, T, \text{In})$ into $Card_P$. Consequently, $\theta(P, Y)$ increases to $\theta(P, Y + T)$ if a result that he verifies $AInfo_{j\,P}$ is valid.

3. 4. 3. 3. 3. A One-way Communication Incentive Model

A one-way communication incentive model (see Figure 3.4.5) is a model that 1) a sending node *s* requests a transferring node *t* to forward a message with *RP* to a receiving node *r*, 2) the node *t* forwards the message with *RP* to the node *r* from the node *s*, 3) the node *r* sends the *RP* to a proxy node *j*, 4) the proxy node *j* sends the *RP* to the node *t*. The model is similar to *Packet Purse Model* [BH00]. However *Packet Purse Model* has a problem that it is difficult to determine the number of *RP* because the number of hop from the node *s* to the node *r* is not clear. For example, DSR protocol [DSR] determines the number of hop before sending a message. On the other hand, AODV protocol [AODV] cannot determine the number of hop before sending a message. For solving the problem, a one-way communication incentive model provides a function by which the node *s* adds *RP* that

composes of an average number of hop and plus something extra to a message and a proxy node j returns the rest of RP to the node s. In addition, this proposed mechanism has message authentication / encryption functions between the node s and r. The study assumes that the node s and r mutual authenticate and they share a symmetric key $CKey_{s\,r}$. For simplifying description, the thesis treats a case of only one transferring node.

1. The node s withdraws $\theta(s, W, \text{Out})$ of W points from $Card_s$. The node s generates Enc_s (a ciphertext is that a plaintext *Message* is encrypted) and the corresponding hash value $Hash_s$ (= H(Enc_s)), and then he sends $Header_s$ (= $ID_s \parallel ID_t \parallel ID_r \parallel Hash_s \parallel \theta(s, W, \text{Out}) \parallel W$), the corresponding $AInfo_{s\,j}$, $\text{AInfo}_{s\,r}$, and Enc_s to the node t.
2. The node t does $W \leftarrow W - 1$ if the output of H(Enc_s) is equal to $Hash_s$. Then the node t sends $Header_t$ (= $ID_s \parallel ID_t \parallel ID_r \parallel Hash_s \parallel \theta(s, W, \text{Out}) \parallel W$), the corresponding $AInfo_{t\,j}$, $AInfo_{s\,j}$, $AInfo_{s\,r}$, and Enc_s to the node r. In case of plural transferring nodes, step 2 is repeated by changing r to t until a next node is the node r.
3. The node r decrypts the encrypted *Message* if the output of H(Enc_s) is equal to $Hash_s$ and a result that he verifies $AInfo_{s\,r}$ is valid. Then the node r sends $Header_r$ (= $ID_s \parallel ID_t \parallel ID_r \parallel Hash_s \parallel \theta(s, W, \text{Out}) \parallel W$), the corresponding $AInfo_{r\,j}$, $AInfo_{s\,j}$, $AInfo_{t\,j}$ to the proxy node j.
4. The proxy node j sends $Header_j$ (= $ID_s \parallel ID_t \parallel ID_r \parallel ID_j \parallel Hash_s \parallel \theta(s, W, \text{Out})$) and the corresponding $AInfo_{j\,M}$ to M if results that he verifies $AInfo_{s\,j}$, $AInfo_{t\,j}$, and $AInfo_{r\,j}$ are valid.
5. M sends $\theta(s, W, \text{Out})$ to C if a result that he verifies $AInfo_{j\,M}$ is valid. Then M receives $\theta(t, W_t, \text{In})$, $\theta(r, W_r, \text{In})$, and $\theta(s, W_s, \text{In})$ from C ($W = W_t + W_r + W_s$.). C does not issue $\theta(t, W_t, \text{In})$, $\theta(r, W_r, \text{In})$, and $\theta(s, W_s, \text{In})$ if $AInfo_{j\,M}$ and $\theta(s, W, \text{Out})$ are invalid. Next, M sends $ID_M \parallel ID_j \parallel \theta(t, W_t, \text{In}) \parallel \theta(r, W_r, \text{In}) \parallel \theta(s, W_s, \text{In})$ and the corresponding $AInfo_{M\,j}$ to the proxy node j.
6. The proxy node j sends $ID_j \parallel ID_t \parallel \theta(t, W_t, \text{In})$, the corresponding $AInfo_{j\,t}$, $ID_j \parallel ID_r \parallel \theta(r, W_r, \text{In})$, the corresponding $AInfo_{j\,r}$, $ID_j \parallel ID_s \parallel \theta(s, W_s, \text{In})$, and the corresponding $AInfo_{j\,s}$ to the node t, r and s respectively if a result that he verifies $AInfo_{M\,j}$ is valid.
7. The node t, r and s input $\theta(t, W_t, \text{In})$, $\theta(r, W_r, \text{In})$, and $\theta(s, W_s, \text{In})$ to $Card_t$, $Card_r$, and $Card_s$ respectively if results that he verifies $AInfo_{j\,t}$, $AInfo_{j\,r}$, and $AInfo_{j\,s}$ are valid. Consequently, RP that each *Card* has increases.

If a transferring node does not forward a message with RP to a next transferring node, the transferring node sends the RP and the corresponding authentication information to M via a proxy node j. Next, M sends the RP to C if the RP and information are valid, and then C transforms the RP to data by which a sending node and the transferring node can input to *Card*. Finally, M sends the data to the sending node and the transferring node via the proxy node j so that this proposed mechanism can prevent RP from being lost.

For improving motivation of nodes to send authentication information to proxy nodes, M should return RP that the nodes spend for sending the information to the nodes.

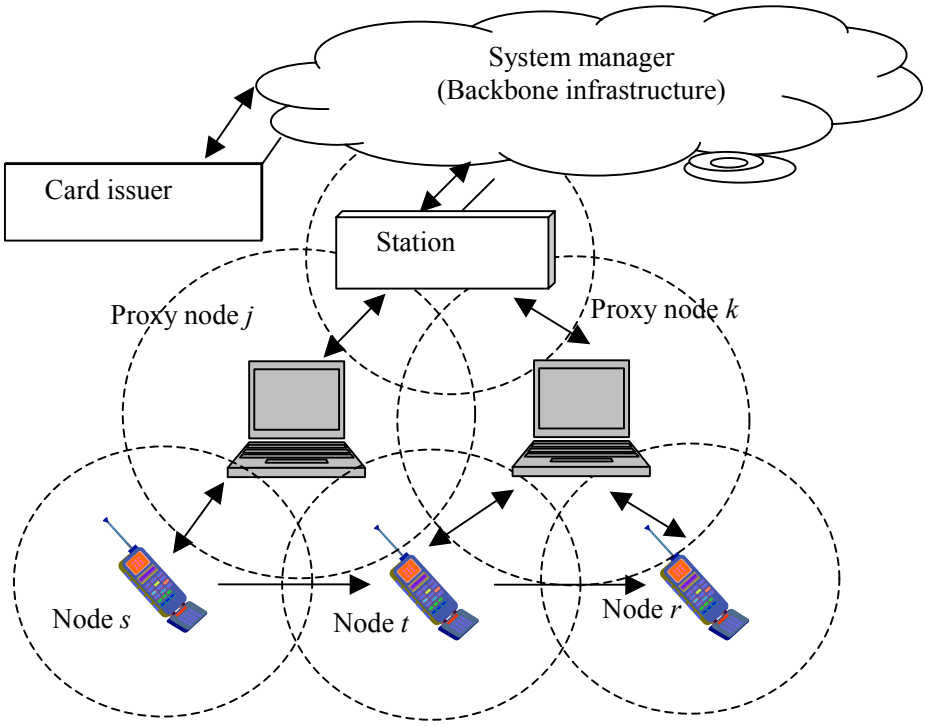


Fig. 3.4.5. A one-way communication incentive model: a node *s* sends a message to a node *r* via a node *t*.

3. 4. 3. 4. Other Consideration

Conclusively a system manager *M* summarizes a dishonest node list of PKI-supporting function and plural authentication information of incentive function via proxy nodes. *M* identifies all the dishonest nodes of two functions and generates a black list of the nodes for improving system security. Next, *M* distributes the black list to all the honest nodes via proxy nodes, and then all the honest nodes reject to connect with all the dishonest nodes of the black list so that all the dishonest nodes are revoked.

In case that a proxy node cannot verify CRL on on-line, the proxy node sends the registered node certificates to *M* when demanding *RP*, and then *M* verifies the certificates from CRL. If invalid nodes exist, *M* notices to the proxy node.

3. 4. 4. Evaluation

3. 4. 4. 1. Confirming Our Design Policy

This subsection verifies to realize our design policy in subsection 3.4.2.2:

1. This proposed mechanism does not request to go through a backbone infrastructure for messages and does not restrict routing methods, because the backbone infrastructure can collect

dishonest node information and authentication information via proxy nodes that PKI realizes.

2. PKI-supporting and incentive functions can share processes (registration with proxy nodes, collection of dishonest node information and authentication information, and black list generation) and the security improves, because of sharing the black lists and distributing the lists via the proxy nodes.

As mentioned above, the proposed mechanism has realized the design policy. In addition, the thesis clarifies as follows:

Services of a PKI-supporting function by which an incentive function needs are mutual authentication, key sharing, and a public key directory service.

Services of an incentive function by which a PKI-supporting function needs are one-way and mutual communication incentive models.

3. 4. 4. 2. Viability of Proposed Mechanism

This subsection verifies that this proposed mechanism satisfies the properties in Section 3.4.2.2.

3. 4. 4. 2. 1. Performance

<Connecability>

This study calculates verification practical probabilities with a PKI-supporting function and without the function. A practical probability (P) is a probability that nodes can execute verification since the nodes can have or obtain CA certificates that is needed for verification. A probability (P_{AB}) of mutual authentication between node A and B composes of a probability (P_{ME}) of mutual authentication between system manager M and proxy node E, a probability (P_{MF}) of mutual authentication between M and proxy node F, a probability (P_{EA}) of mutual authentication between E and A, and a probability (P_{FB}) of mutual authentication between F and B. For simplifying the following discussion, 1) this study presumes that resources of E and F are the same and this thesis calls them "j" so that P_{Mj} (= P_{ME} = P_{MF}), 2) the study presumes that resources of A and B are the same and the thesis calls them "i" so that P_{ji} (= P_{EA} = P_{FB}). In addition, P_{Mj} composes of a probability ($P1_{Mj}$) by which M can execute verification of j's certificate and a probability ($P2_{jM}$) by which j can execute verification of M's certificate. P_{ji} composes of a probability ($P1_{ji}$) by which j can execute verification of i's certificate and a probability ($P2_{ij}$) by which i can execute verification of j's certificate. As mentioned above, P_{AB} is described the following equation:

$$P_{AB} = P_{Mj}^{2} \cdot P_{ji}^{2} = P1_{Mj}^{2} \cdot P2_{jM}^{2} \cdot P1_{ji}^{2} \cdot P2_{ij}^{2} \qquad (3.4.1)$$

From the definition in Section 3.4.2, $P1_{Mj}$ = $P2_{jM}$ = 1. The study expects $P1_{ji}$ = 1 for the following reasons: 1) j has many resources, 2) another proxy node may issue a temporary node certificate to i, and 3) another proxy node may provide CA certificates to j. Therefore, $P_{AB} = P2_{ij}^{2}$

and P_{AB} is based on $P2_{ij}^{2}$.

On the other hand, a verification practical probability without PKI-supporting function is equal to a probability (P_{AB}') by which A can execute verification of B's certificate. P_{AB}' is fixed because of not changing a target node (i.e. B). In a word, $P2_{ij}$ is higher than P_{AB}' since $P2_{ij}$ can select another proxy node and can obtain CA certificates from authenticated node. Thus, the proposed mechanism satisfies *connectability*.

<Efficiency>

This subsection compares the proposed mechanism and the existing efficient scheme [SBHJ03] on cellular networks (see Table 3.4.1). Session establishment processes of the existing scheme and proxy node registration processes of the proposed mechanism are not targets of the comparison because their processes differ in timing of execution. As adjusting the scheme, an output size of one-way hash function is 16 bytes, a size of ID is 4 bytes, a counter is 2 bytes, and RP is 16 bytes. Here, V is the number of transferring nodes, and U is the total number of packets for one message.

Table 3.4.1. Amount comparison of trafic / calculation between our proposed mechanism and the existing scehme [SBHJ03]

Schemes \ Performances		An amount of data for adding to a message when transferrring the message [byte / 1 message]	A total amount of calculation for transferring a message [the number of hash operation / 1 message]
Transfering operation	Our proposed mechanism	$79 + 21V$	$5 + 2V$
	The scheme [SBHJ03]	$22U$	U
Finishing operation from receiving node to system manager	Our proposed mechanism	$77 + 21V$	4
	The scheme [SBHJ03]	$38U$	U

From Table 3.4.1, this proposed mechanism has the advantage if nodes send large messages, and the existing scheme has the advantage if nodes communicate with many hops. The proposed mechanism is more effective than the existing scheme, since network traffic increases year by year and nodes may use stations in case of many hops.

3. 4. 4. 2. 2. Security

Attackers are nodes that purpose as follows:

Illicit obtaintment of RP,

Nonpayment of RP,

Interference with licit obtaintment of *RP*,

Impersonation of nodes and stations, and

Illicit reject of cooperation.

This study presumes that attackers can eavesdrop communications and send any messages, however attackers are difficult to attack *Card* and backbone infrastructure.

<Verifiability>

On proxy node approving and registration with proxy nodes, impersonation and illicit obtainment of secret keys are difficult since nodes mutual authenticate using PKI and then they share a symmetric key using the public key.

On PKI-supporting function, impersonation and illicit obtainment of new secret keys are hard since the nodes share a new symmetric key using the above symmetric key. In addition, a requesting node reports a fact to a system manager *M* if the request and the corresponding provided information are different on public key directory service.

On a mutual communication incentive model, a providing node may not provide requested information when the node receives *RP* form a requesting node, also *RP* by which a providing node receives may be invalid. For preventing the former case, the providing node can report the injustice fact to *M* via a proxy node. For preventing the latter case, a card issuer *C* can find the invalid *RP* when *M* sends the *RP* to *C*.

On a one-way communication incentive model, *M* can find an injustice verifying logs (*Header* and *AInfo*) that are obtained from a sending node, a transferring node, and a receiving node.

Table 3.4.2 shows expected attacks and logs that detect the attacks. "*YES*" means this proposed mechanism can detect an attack, and "*NO*" means the mechanism is difficult to detect an attack. From Table 3.4.2, the proposed mechanism needs to collect logs of a sending node, a transferring node, and a receiving node for preventing all the expected attacks. Moreover, attackers may join existing routes unnecessarily for obtainments of *RP*. However the attacker's motivation is low, because this attack requires costs (e.g. a CPU workload and power consumption) that are equal to valid message forwarding costs and a value of *RP* is not high on the proposed mechanism. For preventing this attack, the proposed mechanism can use the detectable schemes [CBH03][HPJ03] if a value of *RP* is very high.

<Robustness>

Nodes conspiracy does not leak system secret information for breaking a system, since nodes have not the system secret information and a node's private key does not depend on that of another node. Therefore this proposed mechanism has *robustness*.

Table 3.4.2. *AInfo* and *Header* that detect attacks

Attacks \ Logs				*AInfo* and *Header* that detect attacks		
Type	Detail	Injured node	Attacker	Sending node	Transferring node	Receiving node
Illicit obtaintment of *RP*	Attackers demand *RP* for nonexistant messages.	Sending node	Transferring node	*YES*	*YES*	*YES*
			Receiving node	*YES*	*YES*	*NO*
			Transferring node Receiving node	*YES*	*NO*	*NO*
	Attackers demand *RP* for not forwarding messages.	Sending node	Transferring node	*YES*	*YES*	*YES*
			Receiving node			
			Transferring node Receiving node	*YES*	*YES*	*NO*
Interference with licit obtaintment of *RP*	Attackers delete *AInfo* form messages.	Transferring node	Sending node			
			Transferring node	*NO*	*YES*	*NO*
			Receiving node	*NO*	*YES*	*NO*
Nonpayment of *RP*	Attackers add false *RP* to messages.	Transferring node Receiving node	Sending node	Card issuer can find false *RP*.		
Illicit stop of forwarding	Attackers does not forward messages.	Sending node	Transferring node	*NO*	*YES*	*NO*
	Attackers delete *RP* and the correponding AInfo from messages.	Sending node	Transferring node	*NO*	*YES*	*NO*

<Traceability>

This proposed mechanism can trace dishonest nodes because of requiring registration with proxy nodes and proxy node approving. If there are no proxy nodes in a route, the proposed mechanism is difficult to distinguish a dishonest transferring node from a next honest transferring node when the attacking node forges a message on the message forwarding. However the mechanism is deterrent to the attack, since the attacker is detected as a candidate for the attack if the attacker repeats the attack.

3. 4. 4. 4. Comparison of Other schemes

Since this proposed mechanism is only method for providing PKI-supporting and incentive functions simultaneously, this subsection individually compares of the proposed mechanism and existing schemes In addition, comparison targets include a well-known certificate verification scheme [MAMGA00] on a server-client model, because there is no other PKI-supporting scheme for multi-hop cellular networks.

3. 4. 4. 4. 1. Comparison of PKI-supporting Function

From Table 3.4.3, this proposed mechanism is better than existing schemes on multi-hop cellular networks, because the mechanism has all the functions and an ordinary node workload is light.

Table 3.4.3. PKI-supporting funcation comparison between our proposed mechanism and existing schemes

Targes for evaluation / Schemes	Function		Performance		
	Certificate management	Vefication supporting	System	Performer	Workload of ordinary nodes
Proposed mechanism	*YES*	*YES*	Multi-hop cellular network	Privileged node	Light
The scheme [MAMGA00]	*NO*	*YES*	Server-Client model	Server	Light
The scheme [CBH02] The scheme [HBC01]	*YES*	*YES*	Ad-hoc network	Ordinary node	Heavy
The scheme [KZLLZ01] The scheme [YK02] The scheme [ZH99]	*YES*	*NO*	Ad-hoc network	Privileged node	Heavy (Ordinary nodes must connecte to plural privileged nodes)
The scheme [WT01]	*NO*	*YES*	Ad-hoc network	Ordinary node	Heavy

3. 4. 4. 4. 2. Comparison of Incentive Function

From Table 3.4.4, this proposed mechanism is better than existing schemes, because the routing is flexible.

Table 3.4.4. Comparison of incentive funcation between our proposed mechanism and existing schemes

Targes for evaluation / Schemes	System	Depository of *RP*	Principal technology	Routing
Proposed mechanism	Multi-hop cellular network	TRM	PKI and symmetric key cryptography	No limmit
The scheme [BH00] The scheme [BH02]	Ad-hoc network	TRM	TRM	Unknown
The scheme [JHB03] The scheme [SBHJ03]	Restricted multi-hop cellular network	Account	Symmetric key cryptography	Source routing
The scheme [ZYC02]	Ad-hoc network	Unknown	PKI	Source routing
The scheme [BB02] The scheme [MGLB00]	Ad-hoc network		Observation	Source routing

3. 4. 5. Conclusion

This study proposed a new security mechanism that provides an integrated service of PKI-supporting and incentive functions for multi-hop cellular networks. Only this proposed mechanism can provide PKI-supporting and incentive functions simultaneously, moreover the routing is more flexible than that of existing schemes.

Chapter 4. Location Management

4. 1. Introduction

Newly, context awareness services, which use contexts of mobile nodes, are studied actively on ubiquitous computing / networks. Especially, location-based services (LBSs), which use geographical location information of mobile nodes as context, receive mach attention. Services of LBSs include information distribution systems to a specific location, navigation systems for walkers, tracking systems of mobile nodes, and location-based access control systems (e.g. a ticket gate), along with other applications. With diversification of LBSs in the near future, this study expects to increase LBSs that require high security and anonymity. Therefore secure location management technology is major subject of study.

This section indicates two problems of existing location management techniques using communication delay on the target networks, and then proposes two secure location verification schemes for solving the problems. Location verification schemes aim to solve *Location Verification Problem* (a verifier V verifies the fact that a prover P exists in a location L at a time T). For solving *Location Verification Problem*, plural location verification schemes have been proposed. However the schemes have the following problems:

1. The existing schemes are not secure against *Relay Attack*.
2. The existing schemes cannot verify relation distances between plural provers.

By measuring an authentication processing time, the study proposes two schemes: 1) a location verification scheme resistant against relay attack, and 2) plural provers verifiable location verification schemes, as solutions of the problems to Section 4.2, and Section 4.3, respectively.

Section 4.2 describes a secure location verification scheme resistant against *Relay attack*. This subsection shows *Relay attack* that relays communications between a prover and a verifier. The attack can be applied to existing location verification schemes.

Section 4.3 describes location verification schemes to verify locations of plural provers. Here, ordinary location verification schemes cannot treat location relation between two provers. The proposed location verification schemes can verify the location relation since a prover relays a challenge message for another prover.

4. 2. A Location Verification Scheme Resistant Relay Attack

4. 2. 1. Introduction

4. 2. 1. 1. Background

Recently, new mobile networks (mobile IP, ad-hoc / mesh, and ubiquitous) have been studied actively. On the networks, location information of mobile nodes is very important, since the networks provide services depending on real environments using location measurement techniques (e.g. base station-based methods, global positioning system (GPS), and radio frequency identification (RFID)). Above all, this study focuses on location-based services (LBSs).

LBSs are services that depend on a geographical location of a node who requests the LBSs or that of other entity. For instance, there are a service that tracks location of cellular phones and a service that gives some shop and weather information in neighborhood of cellular phones to the phones. Also, the networks may require a node-tracking system, because mobile nodes are moving mostly. The location information provided by the system is important privacy. The privacy should be access-controlled appropriately. In the papers [ITWUSM00][WTTUM96], node-tracking systems for protecting node privacy are proposed. In their systems, a node can publish its own location information to entities that is decided by the node.

4. 2. 1. 2. Location Verification

The papers [BP00][KNF02][KNF03][KNF04][HMYMMA04][OTWFYSK03] introduced enhanced observed time difference (E-OTD), time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and received signal strength (RSS) as location measurement techniques for mobile nodes. These techniques are based on GPS, wire-less LAN systems, and so on. Especially, E-OTD, TOA, and TDOA decide location of nodes using a difference of arrival times of radio waves and ultrasound. However, their techniques are not always secure. For instance, a dishonest node may impersonate another node, a dishonest node may not synchronize a time, and a dishonest node may inform fake location information. This study calls this problem “location verification problem”. In other wards, the problem is that “Can a verifier *V* verify the fact that a prover *P* exists in a location *L* at a time *T*?”

To realize secure LBSs, it must solve the location verification problem. On the problem, it is important to prevent *V* from believing fake location information *L*’ of *P*.

In this thesis, a location is a general term for “Distance, Region, Position and Route (see Figure

4.2.1)". The Distance is a relative distance between *V* and *P*. The Region is a region that *V* can verify the fact that *P* exists. The Position consists of the Distance and a direction from *V* to *P*. And, the study considers the Route consists of the plural Distance information arranged time sequentially.

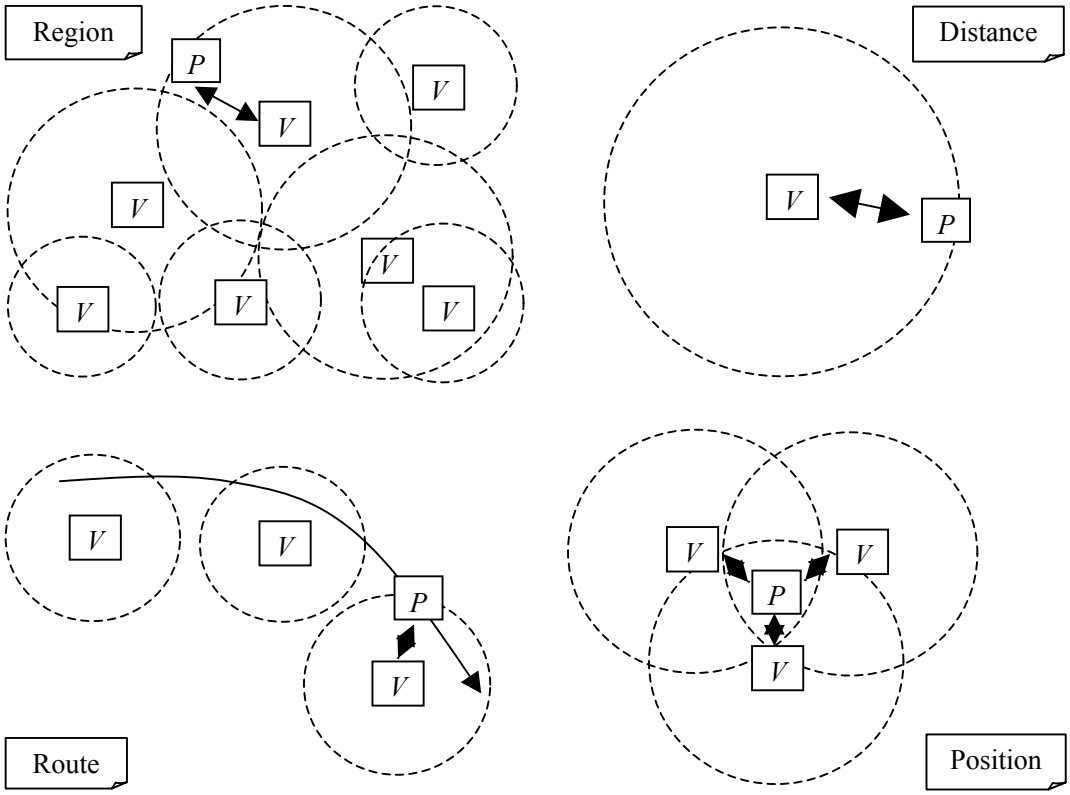


Fig. 4.2.1. Types of location.

4. 2. 1. 3. Related Works

Two approaches were proposed to solve the location verification problem. A first approach [GW98][NNT03] is based on location measurement techniques (e.g. GPS and RFID), and a second approach [BC93][CH04][SSW03][WF03] is based on transmission delay. The papers [GW98] [NNT03] proposed schemes that *V* can obtain location information from *P*, which is a tamper-resistant module included a GPS receiver or an IC tag (the thesis considers that scheme [NNT03] requires tamper-resistance to protect an ID of IC tag). However, an attacker may forge *P*'s location information even if *P* is the tamper-resistant module. For instance, according to emulate / forge / replay GPS signals that are input to a GPS receiver of *P*, the information that *V* obtains may be operated [CH04]. And, according to read the ID of tag using a reader / writer, a clone of *P* may be generated.

The papers [BC93][CH04][SSW03][WF03] proposed location verification schemes using radio frequency (RF) communication delay. The schemes can decide a relative distance and *V* and *P* by measuring a time difference between a time T_C in which *V* sends challenge data to *P* and a time T_R in which *V* receives the corresponding response data from *P*. The schemes have a basis of security in the fact that a speed of RF is equal to one of light in a vacuum. Thus *P* cannot send the response data faster than the speed of light. On the other hand, *V* can verify the fact *P* exists within a distance D (called "Distance verification") since *P* can delay to send the response data. The study thinks that *P* need not to delay, because *V* provides services if *P* exists within a distance D' in general. Moreover, "Region verification and Position verification" are realized if plural verifiers measure distances of *P* by using the distance verification.

The paper [BC93] proposed to use an authentication scheme together with location verification for preventing mafia frauds (i.e. man in the middle attack). Similarly, scheme [WF03] aims for guarding location information of *P* against third parties on an assumption that the *P* is honest. And the scheme resists proxy attack that a third party forges location information of *P* according to set a proxy between *P* and *V*. On the other hand, schemes [CH04][SSW03] aim for preventing dishonest provers and third parties on an assumption that *P* lies to *V* about own location.

On ad-hoc networks, wormhole attacks [CBH03][CHJ04][HPJ03] and a rushing attack [HPJ03] (like a man in the middle attack) were proposed, and these attacks assume that dishonest forward nodes may exist between a sender and a receiver. Also the papers [CBH03][CHJ04][HPJ03][HPJ03] proposed to use location verification as countermeasures against wormhole and rushing attacks.

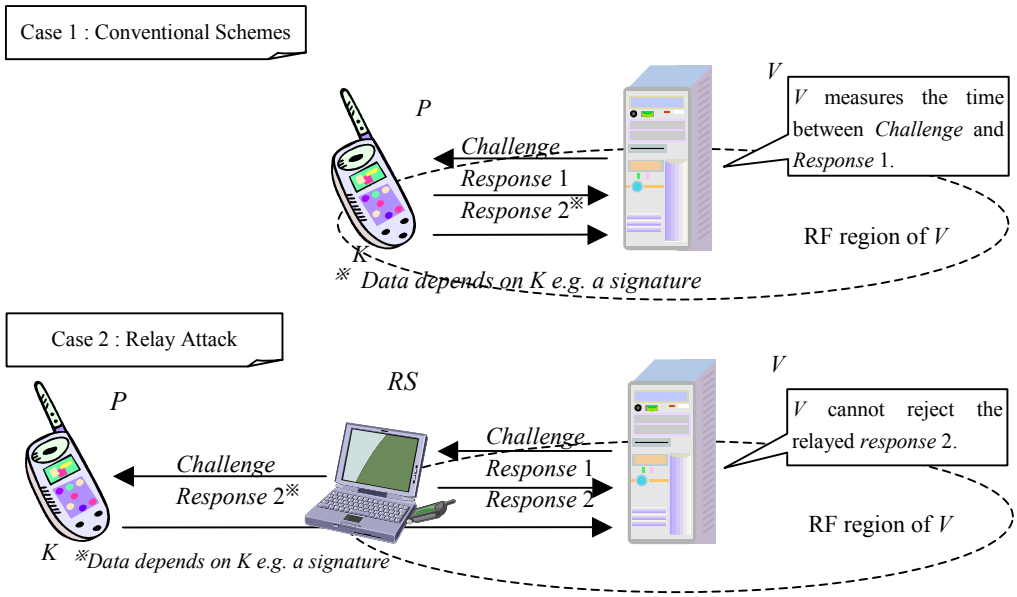


Fig. 4.2.2. Conventional schemes and a relay attack.

In the papers [BC93][CH04], how to authenticate an entity that is a target of location verification are discussed. This problem is that "Can *V* authenticate an identity of *P* when *V* verifies a location of *P*?" The thesis simply calls such this authentication "Authentication" in this thesis. In the paper [SSW03] *V* does not authenticate an identity of *P*. The thesis calls such the case "No authentication". In the paper [WF03], *P* can publish *P*'s ID and location information generated by *V* to a third party selected by *P*. The third party can verify the location information corresponds to the ID. Thus *P* has anonymity against *V*. The thesis calls such this authentication "Authentication with anonymity".

Case 1 of Figure 4.2.2 shows that schemes [BC93][CH04] do not include a delay depending on authentication process. The study expects that their designs aim to improve precision of location verification by excluding the authentication delay from a total delay of the schemes. Moreover, scheme [SSW03] has an acceptable range of delay caused by other processes (e.g. error correction, modulation, and decoding) since the scheme assume a processing time of *P*.

4. 2. 1. 4. Problems of Related works

Related works have at least one of three problems:

1. The study points out a weakness of location verification schemes with authentication [BC93] [CH04][WF03]. A dishonest *P* can force *V* to believe a fake location *L*' of *P* if a relay station *RS* operated by *P* exists between *P* and *V* (the thesis called "Relay attack"). A similar attack is slightly described in the paper [BC93]. Note that the Relay attack differs from Proxy attack [WF03] in that *P* controls *RS*. Case 1 of Figure 4.2.2 shows an outline of schemes [BC93] [CH04][WF03], first *V* measures a difference time between T_C that *V* sends a challenge and T_R that *V* receives the corresponding response 1 for location verification, second *P* generates the corresponding response 2 using own secret key *K* and the challenge and then sends it to *V*, finally *V* receives and verifies the response 2 for authentication. Case 2 of Figure 4.2.2 shows that Relay attack assumes that *P* (with *K*) exists outside a RF range of *V* and *RS* (without *K*) exists within the range. In Relay attack, first *RS* receives a challenge and sends the corresponding response 1, second *RS* relays the challenge to *P* and *P* generates the corresponding response 2 using *K* and the challenge, finally *RS* relays the response 2 to *V* and then *V* verifies the response 2. The first step is that *RS* can respond since *V* does not request a process of *K*. The second step is that *P* can respond since *V* does not measure a transmission time between *P* and *V*. Thus Relay attack can force *V* to accept a location *L*' of *RS* as a location of *P*. On the scheme [WF03], a third party accepts *L*' similarly.
2. The schemes [BC93][CH04][WF03] do not consider a processing delay of *P* (e.g. error correction, modulation, and decoding on communication between *V* and *P*. The study thinks the schemes cannot ignore the processing delay in practical systems.
3. The schemes [BC93][CH04][SSW03][WF03] do not have modularity that *P* can select an

authentication method from “Authentication”, “No authentication”, or “Authentication with anonymity”. The study presumes that the modularity becomes important as diversification of LBSs, since the diverse LBSs request various authentication methods. “Authentication with anonymity” is specially needed for preventing privacy of *P*.

4. 2. 1. 5. Our Results

This thesis proposes a location verification scheme can solve the above-mentioned three problems. A cause Relay attack is effective in the schemes [BC93][CH04][WF03], is that the schemes do not measure a processing time of secret key. Thus this proposed scheme measures the key processing time (see Figure 4.2.3) and the communication processing time. Further the proposed scheme has a flexible framework can *P* can select a method from “Authentication”, “No authentication”, or “Authentication with anonymity” freely.

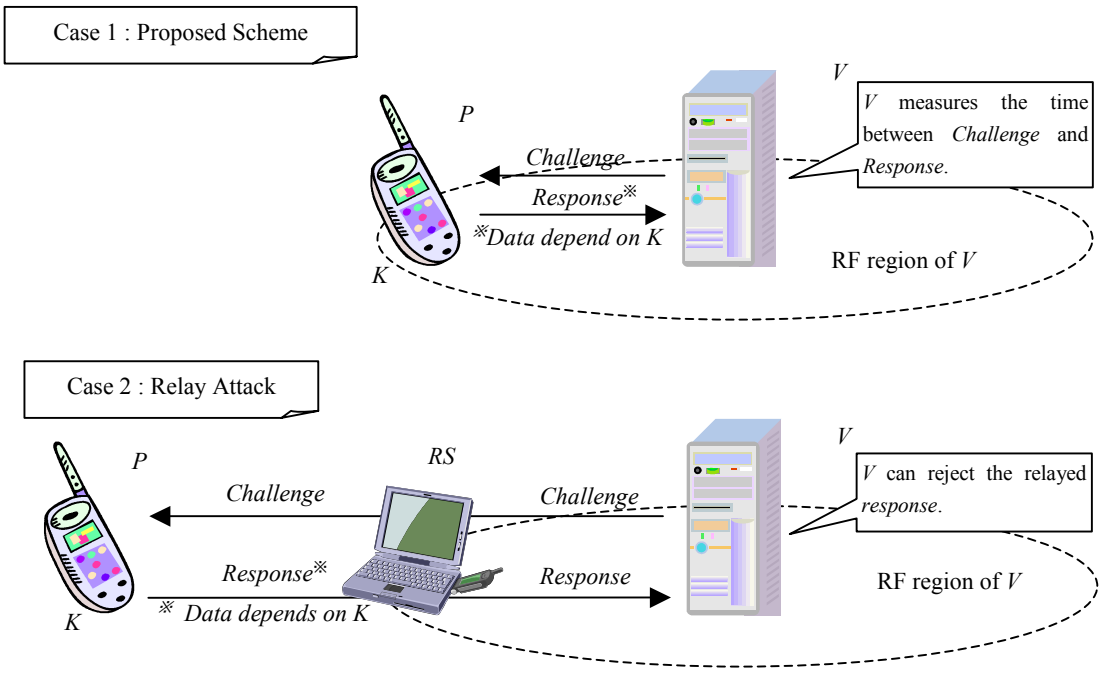


Fig. 4.2.3. Our proposed scheme.

4. 2. 2. Requirements

4. 2. 2. 1. Entities

The proposed scheme consists of the following entities:

Proxy node (*PN*) is a verifier that verifies a location L_i of processing a secret key K_i. *PN* can communicate entities within a distance *r* using RF and can measure a time closely. *PN* is fixed in

a location L_{PN} and detects that *PN* is moved. All entities trust *PN*. Thus *PN* does not operate anything illegal. Due to satisfy the above conditions, the study assumes *PN* has tamper-resistance.

Node *i* (N_i) is a prover that proves own location L_i to *PN*. N_i has a secret key K_i (for symmetric ciphers) or X_i (for asymmetric ciphers and the corresponding public key Y_i) securely and **"*i*"** is ID of *N*. N_i generates a request and a response for a challenge generated by *PN* using own K_i or X_i. Here, it is hard to change the processing time of N_i. Due to satisfy the above conditions, the study assumes N_i has tamper-resistant.

Relay station *i* (RS_i) is an entity that relays communication between *PN* and N_i. RS_i cannot generate a response, since he does not have K_i or X_i. A dishonest N_i controls RS_i.

4. 2. 2. 2. Requirements

This subsection explains four requirements that location verification schemes must satisfy. The thesis only treats distance as location, since distance verification is extended to region verification and position verification.

Distance verifiability

Proposition 1: *While from a time T to a time T + ΔT, a secret key has been calculated within a distance d from a verifier V.*

V can judge that Proposition 1 *is true if V starts observation in T and receives a response until T+ΔT.*

Due to add authentication to distance verifiability, the study guarantees that the secret key and N_i cannot separate, since N_i is a tamper-resistant module and all the entities except N_i do not know the secret key (but *V* knows the key in case of symmetric key cryptography). Also the distance verifiability is "No authentication" if *V* does not know about information of the secret key.

Relay attack-resistance is that a distance verification scheme can resist against Relay attack.

Adaptability is that a distance verification scheme can be adapted to practical systems.

Modularity is that a distance verification scheme has a framework that is easy to change plural authentication methods in regard to a secret key.

4. 2. 2. 3. An Attack Model

A purpose of attacks is to cheat *V* into accepting fake location of N_i. And an attacker is a N_i holder or a third party. Attacks (see Figure 4.2.4) are classified into two types.

Attack on entity:

Tampering is an attack that analyzes and forges *N* or *PN* directly.

Black box attack is an attack that guesses secret information of *N* or *PN* from output of *N* or *PN* for various inputs.

Impersonation is an attack that impersonates against *N* or *PN*.

Attack on communication:

Message forgery is an attack forges messages on communication channels.

Message replay is an attack replays messages on communication channels.

Man in the middle attack is an attack that an attacker lies between N_i and *PN*, and the attacker each impersonates the entities.

Relay attack is an attack that RS_i controlled by N_i relays messages between N_i and *PN*.

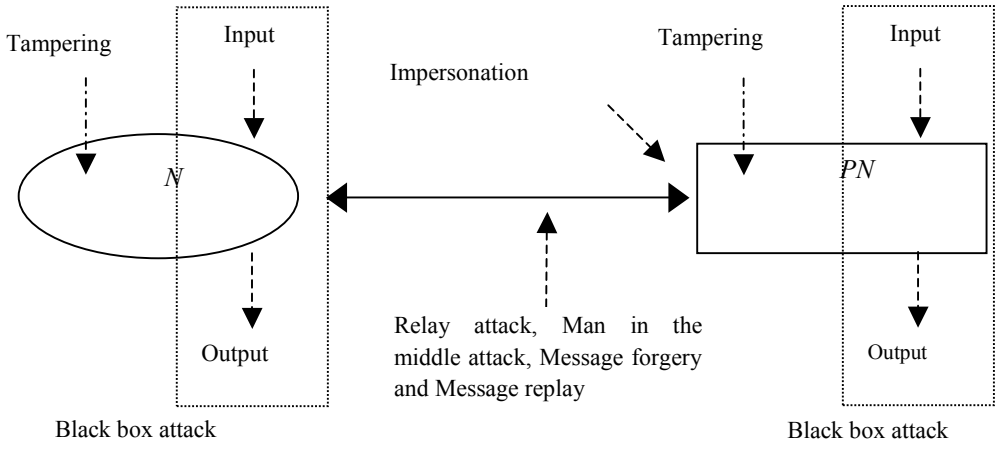


Fig. 4.2.4. Attack model.

4. 2. 2. 4. Data Structure of a Challenge and a Response

In this thesis, a distance *d* between *PN* and N_i is decided by a communication delay between *PN* and N_i. A distance *r* is an available RF distance of *PN*. For preventing Relay attack, *d* should be lager than *r* if RS_i relays communication between *PN* and N_i. Therefore, the study designs challenge data structure that allows communication delay to increase if challenge data is relayed. And this proposed scheme measures communication time and processing time of N_i.

The study assumes that relay delay is always produced if messages is relayed, and the delay increases as follows:

1. an increase of size of relayed communication data,
2. an increase of the number of relays, and
3. an increase of relay distance between N_i and RS_i.

1. For increasing size of relayed communication data, the study should let size of challenge data be enough for detecting the relay delay. However, the relay delay is not large if RS_i separates the challenge data and relays the challenge pieces, and then N_i generates the corresponding response pieces from the challenge pieces and sends RS_i the response pieces (i.e. pipeline processing). Thus

the response data should not be generated if N_i does not have all challenge data. 2. For increasing the number of relays, the study can design a location verification scheme that repeats challenge and response. But, an error of this scheme grows by increasing processing time of N_i. Therefore it is necessary to evaluate the number of repeats in actual systems. 3. The study believes that almost attackers desire to be large a difference between a fake location and a trust one, because the attackers can go to the trust location if the difference is small. Consequently, the thesis does not consider an increase of relay distance.

4. 2. 3. The Proposed Scheme

4. 2. 3. 1. Notation

This thesis defines notation for describing this proposed scheme as follows:

A_Enc(*Y, m*): a secure asymmetric encryption function, which outputs a ciphertext of a message *m* using a public key *Y*. Here, only the corresponding private key *X* can decrypt the ciphertext.

S_Enc(*K, m*): a secure symmetric encryption function, which outputs a ciphertext of a message *m* using a shared key *K*. Here, only *K* can decrypt the ciphertext.

A_Sig(*X, m*) is a secure signature generation function, which outputs a signature of the message *m* using a private key *X*. Here, the validity of the signature is checked using the corresponding public key *Y*.

S_Sig(*K, m*) is a secure MAC (Message Authentication Code) generation function, e.g. HMAC, which outputs a MAC of a message *m* using a shared key *K*. Here, *K* can only check validity of the MAC.

H(*m*) is a secure one-way hash function, which outputs a hash value of a message *m*.

$\mathbf{Mask_Y}$(*R, Y*) is a key mask function that masks a public key *Y* using a random number *R*. It is hard to obtain the public key *Y* from a masked public key *Y*' without *R*.

$\mathbf{Mask_X}$(*R, X*) is a key mask function that masks a secret key *X* using a random number *R*. It is hard to obtain the secret key *X* form a masked secret key *X*' without *R*.

(Q_i, A_i) is a pair of a question and an answer.

Req_i is request data generated by N_i.

Cha_i is challenge data generated by *PN* for N_i.

Res_i is response data generated by N_i.

ReqG is a request data generator, which outputs Req_i from a type of authentication.

ChaG_*C* is a challenge data generator, which outputs Cha_i from a random number and Req_i. *C* indicates a type of authentication as follows: α is "No authentication", β is asymmetric key-based "Authentication", γ is symmetric key-based "Authentication", and ε is "Authentication with anonymity".

ResG_*C* is a response generator, which outputs Res_i from Cha_i and a secret key of N_i. C is an authentication type.

VeriF_*C* is a verification function that outputs "*Accept*" if the input is valid, and a value except "*Accept*" if the input is invalid. C is an authentication type.

$\boldsymbol{t_{Cha}}$ is a time that Cha_i is sent by PN.

$\boldsymbol{t_{Res}}$ is a time that Res_i is received by PN.

$\boldsymbol{T_{Relay}}$ is a relay delay time of an attacker. $T_{Relay} = 0$ if the attacker does not relay Cha_i or Res_i.

$\boldsymbol{T_{Delay}}$ is a processing delay time of N_i that PN allows.

$\boldsymbol{T_{Com}}$ is a communication time between N_i and PN.

$\boldsymbol{\Delta T}$ is a difference time between t_{Res} and t_{Cha}. Here, $\Delta T = t_{Res} - t_{Cha} = T_{Com} + T_{Delay} + T_{Relay}$.

$\boldsymbol{s}$ is a speed of RF.

$\boldsymbol{r}$ is an available RF distance of PN.

4. 2. 3. 2. Framework

This section explains a framework of this proposed scheme. Figure 4.2.5 shows sequence of the proposed scheme.

Step1: N_i generates request data Req_i using the following equation, and then sends Req_i to PN.

$Req_i = \mathrm{ReqG}(Z_i)$

Case 1 "No authentication": $Z_i \leftarrow i \parallel R_i$. Here, "||" indicates concatenation of data, and $B \leftarrow A$ means that A is substituted into B.

Case 2 Symmetric key-based "Authentication": $Z_i \leftarrow i$.

Case 3 Asymmetric key-based "Authentication": $Z_i \leftarrow Y_i$.

Case 4 "Authentication with anonymity": $Z_i \leftarrow Y_i$' (is output of $\mathrm{Mask_Y}(R_i, Y_i)$).

Step2: PN receives Req_i, and then obtains Z_i from Req_i. PN decides a type of authentication requested by N_i from Z_i. PN sends "*Reject*" to N_i if PN does not accept the type, and then PN stops this protocol.

Step3: PN selects a random number R_{PN}, and then generates challenge data Cha_i (includes a question) and the corresponding answer using the following equation. Finally, PN sends Cha_i to N_i and then stores a time of sending Cha_i into t_{Cha}. Note that PN must send Cha_i to N_i in order from MSB (Most Significant Byte).

$(Cha_i, A_i) = \mathrm{ChaG\_}C(R_{PN}, z_i)$

$z_i \leftarrow R_i$ if Z_i is equal to $i \parallel R_i$.

$z_i \leftarrow K_i$, PN selects the corresponding shared key K_i from i if Z_i is equal to i.

$z_i \leftarrow Y_i$ if Z_i is equal to Y_i.

$z_i \leftarrow Y_i$' if Z_i is equal to Y_i'.

Step4: N_i receives Cha_i, and then generates response data Res_i (includes the corresponding answer) using the following equation. Finally, N_i sends Res_i to PN.

Res_i = ResG_C(Cha_i, ζ_i)

$\zeta_i \leftarrow R_i$ if Z_i is equal to $i \parallel R_i$.

$\zeta_i \leftarrow K_i$ if Z_i is equal to i.

$\zeta_i \leftarrow X_i$ if Z_i is equal to Y_i.

$\zeta_i \leftarrow X_i$' (is output of $\text{Mask}_X(R_i, X_i)$) if Z_i is equal to Y'.

Step5: PN receives Res_i, and then stores a time of receiving Res_i into t_{Res}. Next, PN calculates ΔT (= t_{Res} - t_{Cha}), and PN verifies Res_i using the following equation,

VeriF_C(A_i, Res_i, z_i, r, s, T_{Delay}, ΔT).

Finally, PN calculates d (= ($\Delta T - T_{Delay}$) / 2s), and then accepts d if output of VeriF_C is equal to "*Accept*".

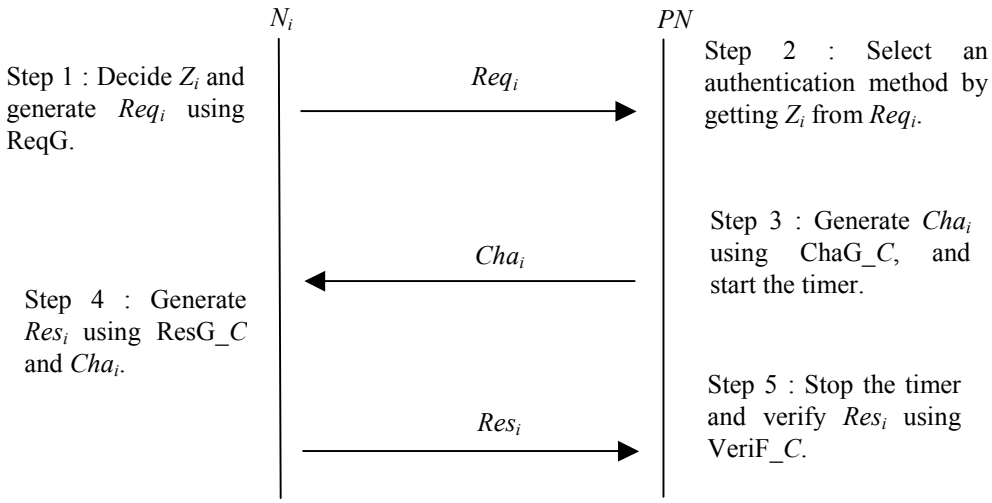


Fig. 4.2.5. Sequence of our proposed scheme.

4. 2. 3. 2. Requirements of a Function and Generators

This subsection describes requirements and details of the above-mentioned function and generators as follows:

ReqG can generate Req_i that gives Z_i to PN.

ChaG_*C* can generate Cha_i that satisfies as follows:

- Cha_i includes an answer A_i.
- A_i can be only obtained from the whole Q_i.
- It is hard to execute processes to generate A_i (or a signed A_i) until obtaining the whole Q_i.
- A_i (or a signed A_i) can be only calculated by ζ_i correspond to z_i.
- Sizes of A_i and the corresponding element A_{i_j} are enough for preventing brute-force attacks.
- CS_i (a size of Cha_i) is enough for causing T_{Relay}. If Cha_i is relayed, T_{Relay} satisfies the following inequality

$$((T_{Com} + T_{Relay} - T_{Delay}) / 2s) > r \qquad (4.2.1)$$

ChaG_β1 : Asymmetric key-based "Authentication" 1

$Cha_i = Q_i$

$= Q_{i\_n} \| Q_{i\_n\text{-}1} \| Q_{i\_n\text{-}2} \| \dots \| Q_{i\_j} \| \dots \| Q_{i\_3} \| Q_{i\_2} \| Q_{i\_1}$

A construction of Q_i in Cha_i is,

Q_{i_1} = A_Enc(Y_i, A_{i_1}), and
Q_{i_j} = S_Enc($A_{i\_j\text{-}1}$, A_{i_j}) ($j \geq 2$), or

A_Enc($Y_{i\_j\text{-}1}$, A_{i_j}) ($j \geq 2$, A_{i_j} and Y_{i_j} are a pair of a private key and the corresponding public key respectively).
In case of $2 \leq n$, Q_i is a multiplex construction. And, in case of $n = 1$, Q_i is a simple construction.
A construction of the corresponding A_i is as follows:

$A_i = A_{i\_n} \| A_{i\_n\text{-}1} \| A_{i\_n\text{-}2} \| \dots \| A_{i\_j} \| \dots \| A_{i\_3} \| A_{i\_2} \| A_{i\_1}$.

ChaG_β2 : Asymmetric key-based "Authentication" 2

$Cha_i = Q_i$

$= Q_{i\_n} \| Q_{i\_n\text{-}1} \| Q_{i\_n\text{-}2} \| \dots \| Q_{i\_j} \| \dots \| Q_{i\_2} \| Q_{i\_1} \| Q_{i\_0}$

$Q_{i\_0} = \Omega_i$ (is a sequence that sets integers from 1 to n at random). In case of $n = 1$, Q_{i_0} is not included in Q_i, for instance, $\Omega_i = 2 \| 5 \| \text{n} \| \dots \| 9 \| n\text{-}3$. Q_i' is a result of that Q_i is re-arranged according to Ω, for example, $Q_i' = Q_{i\_2} \| Q_{i\_5} \| Q_{i\_n} \| \dots \| Q_{i\_9} \| Q_{i\_n\text{-}3}$.

A construction of the corresponding A_i is as follows:

Signed A_i = A_Sig(X_i, H(Q_i')).

Note that Q_i is requested to re-set to Q_i' if $2 \leq n$, also Q_i' is equal to Q_i if $n = 1$.

ChaG_γ1 : Symmetric key-based "Authentication" 1

A construction of Cha_i and the corresponding A_i is same with ChaG_β1. A construction of Q_{i_j} is as follows:

Q_{i_1} = S_Enc(K_i, A_{i_1}), and

Q_{i_j} = S_Enc($A_{i\_j\text{-}1}$, A_{i_j}) ($j \geq 2$).

ChaG_γ2 : Symmetric key-based "Authentication" 2

A construction of Cha_i and the corresponding A_i is same with ChaG_β2. A construction of A_i is as follows:

Signed A_i = S_Sig(K_i, Q_i')

ChaG_α : "No authentication"

A construction of Cha_i and Res_i is same with Symmetric key-based "Authentication" 1 and 2, however it must use R_i instead of K_i as secret key. By the way, it is no information that *PN* traces N_i if R_i is changed at every time. On the other hand, *PN* can distinguish N_i (but he cannot trace N_i) if R_i is same at every time.

ChaG_ε : "Authentication with anonymity"

The thesis proposes discrete logarithm problem (DLP)-based scheme for ChaG_β2. DLP is defined over finite cyclic groups, including subgroups of Jacobians of elliptic curves, and so on. Thus DLP is computationally hard to solve. The section explains the proposed scheme over a prime field $\boldsymbol{Z}_p$. Here, R_i and X_i are in $\boldsymbol{Z}_q$. Also the study uses as follows:

p is a large prime number,

q: is a large prime number such that $q \mid p$-1,

g is a q^{th} root of unity over $\boldsymbol{Z}_p$,

(X_{PN}, Y_{PN}) is a pair of a private key and the corresponding public key for *PN*,

$\boldsymbol{ID_{PN}}$ is ID of *PN*,

$\boldsymbol{Cert_i}$ is a public key certificate of Y_i (= $g^{X_i} \bmod p$),

Y' = $\text{Mask}_Y(R_i, Y_i) = Y_i^{R_i} \bmod p = g^{X_i \cdot R_i} \bmod p$, and

X' = $\text{Mask}_X(R_i, X_i) = R_i \cdot X_i \bmod q$.

Step a): On the above parameters and functions, it executes the proposed scheme (Asymmetric key-based "Authentication" 2).

Step b): On Step5 of the proposed scheme, *PN* sends a location certificate $InfoL_i$ to N_i if output of VeriF_ε is "*Accept*". $InfoL_i$ is output of A_Sig(X_{PN}, ID_{PN} || t_{Cha} || t_{Res} || Y_i').

Step c): N_i receives $InfoL_i$, and then publishes $InfoL_i$, R_i, $Cert_i$, and $Cert_{PN}$ to a third party that N_i selects.

Step d): The third party verifies signatures of $Cert_{PN}$, $InfoL_i$, and $Cert_i$, and then obtains Y_i' and Y_i from $InfoL_i$ and $Cert_i$ respectively if all the signatures are valid. Next, he verifies that Y_i' and the output of $\text{Mask}_Y(R_i, Y_i)$ are same, and then accepts ID_{PN}, t_{Cha} and t_{Res} if the result is valid.

ResG_*C* can generate Res_i includes A_i, using Cha_i and ζ_i.

VeriF_*C* outputs "*Accept*" if input values (A_i, Res_i, z_i, r, s, T_{Delay}, ΔT) are valid, or outputs a value except "*Accept*" if the input values are invalid. *Accept* means to satisfy $r \geq d$ (= ($\Delta T - T_{Delay}$) / $2s$) > 0 and the following conditions:

in cases of VeriF_β1 and VeriF_γ1, A_i is equal to A_i' that is obtained from Res_i,

in case of VeriF_β2, A_i is equal to A_i' that is obtained from Res_i, and a signature of Signed A' that is obtained from Res_i, is valid,

in case of VeriF_γ2, A_i is equal to A_i' that is obtained from Res_i, and a MAC of Signed A' is valid,

in case of VeriF_α, VeriF_α is same with VeriF_γ1 or VeriF_γ2, however it must use R_i instead of K_i as secret key, or

in case of VeriF_ε, VeriF_ε is same with VeriF_β2.

4. 2. 3. 3. Viability of Our Proposed Scheme

This proposed scheme satisfies our all requirements.

Distance Verifiability: Section 4.2.4 explains security of this requirement. This section considers the realization of this proposed scheme. The proposed scheme requires a precision of 3.333… [nsec] per 1 [m], and the approved processing delay is a maximum of 0.333… [nsec] if an error range is within 10 [cm]. In the paper [MK03], a PC with improved software has a time precision of 1 [μsec], and a PC with improved hardware (e.g. a high-frequency crystal oscillator) has a time precision of 34 [ns]. In short, the PC with improved hardware can measure a distance by about 10 [m]. Moreover, the paper [TMIK05] proposed a special time-stamp hardware has a time precision of 8 [ns] and therefore a PC with the hardware can measure a distance by 2.4 [m]. On the other hand, the proposed scheme is a kind of TOA and TDOA. Here, the paper [OTWFYSK03] proposed a system that obtains location by 1 - 4 [m] of a wireless LAN node using TDOA and plural access points (i.e. verifiers). Moreover, the papers [KNF03][HMYMMA04] show to be able to simulate an influence of multipass fading. Therefore the study thinks to be able to realize this proposed scheme that measures a distance by 1 - 10 [m].

Relay attack-resistance: Section 4.2.4 explains this requirement on.

Adoptability: this proposed scheme assumes V knows a processing delay of P, and V considers the delay when V verifies a location of P. Thus the proposed scheme satisfies "Adoptability".

Modularity: the proposed scheme is that a node can easily select one of six authentication methods. Therefore, the scheme satisfies "Modularity".

Next, the study compares between the proposed scheme and other schemes on a point of view of the requirements. From Table 4.2.1, only the proposed scheme satisfies all the requirements. Specially, all other schemes do not satisfy "Relay attack-resistance" and "Modularity". The study thinks that a location verification scheme should satisfy "Relay attack-resistance" due to use in

insecure environments. Also some systems may not require “Modularity”, but the study expects that “Modularity” becomes essential since a requirement of protecting privacy is rising recently.

The proposed scheme requires a tamper-resistant module due to satisfy “Adoptability”. *V* can fix a key processing and communication processing delays (e.g. modulations, error corrections, and decoding) of a tamper-resistant module N_i. On the other hand, the schemes [BC93][CH04][SSW03] do not require a tamper-resistant module since the schemes do not measure a key processing delay. However, a RF protocol requires communication processing in reality. Thus their schemes cannot ignore the processing delays. The study supposes that most location verification schemes need a tamper-resistant module. As mentioned above, the study thinks that this proposed scheme is more excellent than other schemes.

Table 4.2.1. Comparison between our proposed scheme and other schemes

Security \ Scheme	Our proposed scheme	The scheme [BC93]	The scheme [SSW03]	The scheme [WF03]	The schemes [CH04]	
					VM[1]	VTDOA[2]
Distance verifiability	***Yes***	***Yes***		***Yes***	***Yes***	***Yes***
Relay attack-resistance	***Yes***	*No*		*No*	*No*	*No*

Performance \ Scheme	Our proposed scheme	The scheme [BC93]	The scheme [SSW03]	The scheme [WF03]	The scheme [CH04]	
					VM[1]	VTDOA[2]
Adoptability	***Yes***	***Yes***	***Yes***	*No*	*No*	*No*
Modularity	***Yes***	*No*		*No*	*No*	*No*
No Authentication	***Yes***	***Yes***	***Yes***	*No*	*No*	*No*
Asymmetric Key Based Authentication	***Yes***	***Yes***		***Yes***	***Yes***	***Yes***
Symmetric Key Based Authentication	***Yes***	*No*		*No*	*No*	*No*
Authentication with Anonymity	***Yes***	*No*		***Yes***	*No*	*No*

[1] Verifiable Multilateration, [2] Verifiable Time Difference Of Arrival

4. 2. 4. Security Analysis

4. 2. 4. 1. Basic Security of Our Proposed Scheme

This proposed scheme satisfies “Distance verifiability” and “Relay attack-resistance”, based on the following assumptions:

Assumption 1: *PN* knows a delay time T_{Delay} that consists of a secret key processing time N_i needs to calculate A_i (or a signed A_i) and a communication processing time of N_i. Attackers are hard to change T_{Delay}. The proposed scheme requires tamper-resistance to N_i for satisfying this assumption.

Assumption 2: Attackers cannot transmit data faster than *s*. Note that this assumption is reasonable if s is almost equal to a speed of light in a vacuum.

Assumption 3: A relay delay of time T_{Relay} (> 0) occurs if communication between *PN* and N_i is relayed and T_{Relay} satisfies an inequality $((T_{Com} + T_{Relay} - T_{Delay}) / 2s) > r$. Note that the study has designed a challenge and a response of this proposed scheme that satisfy this assumption.

Assumption 4: RS_i and third parties cannot know a secret key of N_i. Note that the study assumes A_Enc(*Y*, *m*), S_Enc(*K*, *m*), A_Sig(*X*, *m*), S_Sig(*K*, *m*) and H(*m*) are secure, and N_i is a tamper-resistant module.

The proposed scheme decides that N_i exists within a distance *d* (= $(T_{Com} + T_{Relay} - T_{Delay}) / 2s$) from *PN* if output of VeriF_*C* is "*Accept*". Here, *d* satisfies $r \geq d > 0$. From Assumption 1 T_{Delay} is not changed by attackers. From Assumption 2 T_{Com} is not decreased by attackers. From Assumption 3 T_{Relay} satisfies $((T_{Com} + T_{Relay} - T_{Delay}) / 2s) > r$ if communication between *PN* and N_i is relayed. From Assumption 4 only N_i can calculate A_i (or Singed A_i). Therefore VeriF_*C* is secure so that N_i exists within d from *PN* if output of VeriF_*C* is "*Accept*".

4. 2. 4. 2. Security Parameters and Anonymity

This study discusses parameters and anonymity of the proposed scheme.

A size of challenge data Cha_i is a security parameter that decides a relay delay. In practical manner, the study needs to consider about a transmission speed of RF communication systems and a relay speed of RS_i. Here, CS_i means a size of Cha_i, E_i [bit / sec] means a transmission speed, and F_i [bit / sec] means a relay speed. The study describes T_{Com} and T_{Relay} using E_i and F_i as follows: $T_{Com} = CS_i / E_i$, and $T_{Relay} = CS_i / F_i$, also the equality (1) is re-described as follows:

$$((CS_i / F_i + CS_i / E_i - T_{Delay}) / 2s) > r.$$

In a ward, the study can decide CS_i according to expect F_i.

A size of answer A_i and the corresponding element A_{i_j} requires a size that resists against Brute-force attack.

A security of "Authentication with anonymity": *V* cannot identify N_i since *V* does not know a public key of N_i, if $\text{Mask}_Y(R, Y)$, $\text{Mask}_X(R, X)$, and A_Sig(*X*, *m*) are secure and DLP is computationally hard. On the other hands, a third party selected by N_i can know the public key using the random number *R*.

4. 2. 4. 3. Other Attacks-Resistance

This subsection evaluates attacks-resistance of this proposed scheme.

Tampering is not available to a tamper-resistant N_i and PN. Thus attackers cannot obtain secrets of N_i and PN.

Black box attack is not available since the study presumes to use secure A_Enc(Y, m), S_Enc(K, m), A_Sig(X, m), S_Sig(K, m) and H(m) for generating the response. Thus it is difficult to obtain the key from the response.

Impersonation is not available to PN since PN authenticates N_i using secure A_Enc(Y, m), S_Enc(K, m), A_Sig(X, m), S_Sig(K, m), and H(m).

Message forgery is detected if the proposed scheme adds plural MAC to messages.

Message replay is detected if the proposed scheme includes timestamps or sequential numbers into messages.

Man in the middle attack is detected like relay attack.

4. 2. 5. A Variety of Location Verification Schemes

4. 2. 5. 1. "Region, Position, and Route verification"

Alike the schemes [CH04][SSW03], this proposed scheme can extend to "Region and Position verification". Moreover, the study proposes "Route verification" that PN can verify moving routes of N_i. N_i requests "Distance verification" per a regular time. Consequently, the results of verifications arranged time-sequentially mean the routes of N_i if N_i moves.

4. 2. 5. 2. MAC and Timestamps

From Section 4.2.4, the proposed scheme supposes to use MAC and timestamp to prevent message forgery and message replay. However a precision of location verification becomes worse if ΔT includes transmission and processing time of their data. Thus, a location verification scheme should send a MAC and a timestamp for a challenge before sending the challenge, and verify the MAC and timestamp after sending the corresponding response.

4. 2. 5. 3. Repeat of a Challenge and a Response

On the proposed scheme, from Step3 - Step5 in Section 4.2.3 can be repeated at B ($1 \leq h \leq B$) times. Here, B is a security parameter depending on T_{Relay}. An increase rate of T_{Relay} per CS_{i_h} (is a size of challenge Cha_{i_h} on h times) on the repeating case is lager than one on a non-repeating case if T_{Relay} includes an overhead time that does not rely on a size of relayed data. In the repeating case, the study thinks that a total size of challenge ($B \cdot CS_{i\_h}$) on the repeating case may be smaller than one on the non-repeating case. But an error range of d may extend by increasing an overhead of processing

responses.

4. 2. 5. 4. How to Implement a Proxy Node and a Node

Implementations of *PN* are divided into an independent type and a centralization type. *PN* acts alone on the independent type. A privileged center controls plural *PN*s on the centralization type. On implementations of N_i, the study thinks that specific countermeasures (e.g. [KIAM00][KAM01]) of side-channel attacks (e.g. DPA [KJJ99], timing attacks [K96]) can use to fix communication and key processing delay of N_i, since the countermeasures fix calculation time to maximum using redundant calculations when the time is not maximum.

4. 2. 5. 5. How to Minimize a Key Processing Delay

For keeping an error range depending on a key processing delay to a minimum, this subsection shows that how to minimize the key processing delay by using one-time pad as symmetric key on Symmetric key-based "Authentication" 1 and 2. First, let S_Enc(K, m) and S_Sig(K, m) be a simple operation (e.g. XOR) between inputs for the functions. Second, *PN* and N_i share a secure pseudo random generator (PRG) and a master key K_i. Third, N_i generates w-th one-time key K_{i_w} (that is inputted in the functions) using the PRG before generating a request, and stores the key secretly. Fourth, a question Q_i is constructed on $n = 1$. As above-mentioned conditions, processing cost of Symmetric key-based "Authentication" 1 and 2 are minimal.

4. 2. 6. Conclusion

This thesis proposed a location verification scheme that resists against "Relay attack". Moreover, this proposed scheme has two properties: the scheme has a framework that a user can easily select one from plural authentication methods, and the scheme considers key processing and communication processing delays.

4. 3. Plural Provers Verifiable Location Verification Schemes

4. 3. 1. Introduction

4. 3. 1. 1. Background

Recently, new mobile networks (such as mobile IP, ad-hoc / mesh, and ubiquitous) have been studied actively. On the networks, real environment information of mobile nodes is very important, since the networks provide services that depend on the real environment information (e.g. a location). For instance, location information is obtained by using global positioning system (GPS) and radio frequency identification (RFID). In generally, it calls such the services "location base services (LBSs)".

LBSs are services that depend on a geographical location of a node who requests the LBSs or that of other entity. For example, there are a service that tracks locations of cellular phones, a service that gives shop and weather information in neighborhood of cellular phones to the phones, and a service to certificate a producing center.

To realize LBSs, this study can use location measurement systems based on GPS, wire-less LAN systems, and so on. Especially, location measurement systems using sensor networks [VN04] [HMYMMA04] and RFID [NNT03] are studied actively now. These systems include location measurement methods that the papers [BP00][KNF02][KNF03][KNF04][HMYMMA04] [OTWFYSK03] introduce, for instance, enhanced observed time difference (E-OTD), time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and received signal strength (RSS). This thesis focuses on TOA and TDOA using radio waves. However, their techniques are not always secure. For instance, a dishonest node may impersonate another node, a dishonest node may not synchronize a time, and a dishonest node may inform fake location information. The thesis calls this problem "location verification problem".

4. 3. 1. 2. Related Works

Ordinary location measurement techniques are not always secure since mobile and sensor nodes may provide fake location information. This problem is that "Can a verifier V verify a fact that a prover P exists in a location L at a time T?" To realize secure LBSs, it is important to solve the problem.

For solving location verification problem, the papers [AM05-1][BC93][CH04][SSW03][WF03] introduced location verification schemes using transmission delay of radio frequency (RF). The

schemes can decide a relative distance d between V and P by measuring a time difference between a time T_C in which V sends a challenge message to P and a time T_R in which V receives the corresponding response message from P. The schemes have a basis of security in the fact that a speed of RF is equal to one of light. Thus V can verify the fact P exists within a distance d because P cannot send the response message faster than the speed. However, V only can judge a fact that P exists within d from V because of P can delay to send the response.

In particular, it is practical that the schemes [AM05-1][SSW03] have a design to consider a processing delay of P (e.g. error correction, modulation, and decoding on communication between V and P) and resistance against relay attack [AM05-1]. Note that the scheme [SSW03] cannot authenticate an identity (ID) of P. The relay attack [SSW03] is an attack that a dishonest P can force V to believe a fake location L' of P if a relay station RS operated by P exists between P and V, and RS relays messages that V authenticates P.

On the other hand, the schemes [BC93][CH04] can authenticate P. This problem is that "Can V authenticate an identity of P when V verifies a location of P?" Above all, the scheme [AM05-1] has modularity that P can select the following four authentication methods:

"No authentication" is that V does not authenticate an ID of P such as the scheme [SSW03].

Symmetric key-based "Authentication" is that V authenticates an ID of P using symmetric key cryptography.

Asymmetric key-based "Authentication" is that V authenticates an ID of P using asymmetric key cryptography such as the scheme [BC93][CH04].

"Authentication with anonymity" is that V cannot authenticate an ID of P but a third party who is selected by P can know the ID and location of P such as the schemes [AM05-1][WF03]. Note that the realization methods differ from the scheme [AM05-1] to the scheme [WF03].

4. 3. 1. 3. Problems of Related Works

In future, this study expects appearance of new LBSs, which requires proving location relation between plural provers as LBSs may diversify. As such the LBSs, the study thinks an entrance control in case of that only plural parties can enter a store, a service to certificate a fact that plural parties have met for a legal purpose, and so on.

This problem is that "Can V verify a fact that a distance between a prover P_1 and a prover P_2 is within distance d at time?" This thesis calls this problem "Location verification problem between two points". For distinguishing P_1 from P_2 V needs to authenticate their IDs. Existing location verification schemes [AM05-1][BC93][CH04][WF03] consider improvement of verification precision and extension of verification range by using plural verifiers, however the schemes does not consider to prove location relation between plural provers.

This thesis introduces a few techniques for proving a fact that plural entities exist in at the same

time. On RFID systems, Juels proposes “yoking-proof”[J04] that proves a fact that an IC tag reader reads two IC tags simultaneously. This is a scheme that their two IC tags corporate to generate one signature via the reader. For example, the scheme is used for a medical prescription and the corresponding medicine must be forwarded simultaneously. Moreover, on the paper [SS04] Saito and Sakurai extend the scheme [J04] to more than two tags, and improve the resistant against message replay attacks. Their schemes have a timeout assumption that the protocol stops when the protocol does not complete within a given period of time. The assumption purposes to improve the security, also an IC tag reader has a timeout function in general. However, the time measurement of their schemes does not aim to verify locations of IC tags. Therefore, it is difficult that the schemes [J04][SS04] can prove location relation between IC tags. By a relay attack [AM05-1], the schemes may prove that tags exist within the RF range of the reader relaying communication between the reader and the tags, even if the tags do not really exist within the RF range.

Next, the thesis introduces a scheme [FA04] proposed by Frikken and Atallah that two nodes can calculate a distance between the nodes without knowing mutual locations. This scheme differs from location verification schemes in the purpose since the scheme assumes that a node knows own location beforehand.

4. 3. 1. 4. Our Results

This thesis proposes new location verification schemes that can prove location relation between two provers, and among n (≥ 2) provers (the thesis called “location verification scheme between two points, and location verification scheme among n points”, respectively). To realize the proposed schemes, the study has the following idea: a prover P_1 relays a challenge message that V sends to a prover P_2, also P_2 relays another challenge message that V sends to P_1, when V verifies location relation between P_1 and P_2. From a difference between the above result and results that V respectively verifies locations of P_1 and P_2 without relaying, V can calculate the location relation. The study realizes this proposed schemes extending the location verification scheme [AM05-1] since the scheme considers relaying massages.

In this thesis, a term “location” is includes “distance, region, position, and route”, the same as Section 4.2. The study focuses on “distance” as “location” in this paper, since distance verification can be extended to region, position and route verification.

4. 3. 2. Requirements

4. 3. 2. 1. Entities

This proposed schemes consists of the following entities:

Proxy node (*PN*) is a verifier V that verifies a location L_i of processing a secret key K_i (or X_i). *PN*

can communicate entities within a distance r using RF and can measure a time of the communication closely. PN is fixed in a location L_{PN} and detects a fact that PN is moved, or PN can move and obtain L_{PN} securely. All entities trust PN. Thus PN does not operate anything illegal. Due to satisfy the above conditions, the study assumes PN has tamper-resistance.

Node *i* (N_i) is a prover P that proves own location L_i to PN. N_i has a secret key K_i (for symmetric ciphers) or X_i (for asymmetric ciphers and the corresponding public key Y_i) securely and i ($1 \leq i \leq n$, $n \geq 2$) is ID of N_i. N_i generates a request message and a response message for a challenge message generated by PN using own K_i or X_i. Also, N_i can relay a challenge message or a response message for another node N_j ($i \neq j$). Here, it is hard to change a relaying time and processing time of N_i. Due to satisfy the above conditions, the study assumes N_i is a tamper-resistant module. Moreover, an honest N_i does not move during V is verifying.

Relay station *i* (RS_i) is an entity that relays communication between PN and N_i. RS_i cannot generate a response message since he does not have K_i (or X_i). A dishonest N_i controls RS_i.

4. 3. 2. 1. Requirements

This subsection explains requirements that location verification schemes between two points (or among n points) must satisfy.

Distance verifiability between two points

Proposition 1: *While from a time T to a time $T + \Delta T$, a secret key i has been calculated within a distance d_i from a verifier V, a secret key j has been calculated within a distance d_j from the verifier V, and the key i and key j ($i \neq j$) have been calculated within a distance $d_{\{i, j\}}$.*

V can judge that Proposition 1 *is true or false if V starts observation in T and receives all responses until $T + \Delta T$.*

Since the distance verifiability between two points includes entity authentication, the study guarantees that the secret key i and N_i cannot separate, because N_i is a tamper-resistant module and all the entities except N_i do not know the secret key i (but V knows the key i in case of symmetric key cryptography).

Distance verifiability among n points

Proposition 2: *While from a time T to a time $T + \Delta T$, on a set $U = \{1, \ldots, n\ (\geq 2)\}$ of secret key IDs, a secret key* 1 *has been calculated within a distance d_i from a verifier V, a secret key n has been calculated within a distance d_j from the verifier V, and the key pair $W = \{(i, j) \mid i \in U, j \in U, i \neq j\}$ has been calculated within a distance $d_{\{i, j\} \in W}$.*

V can judge that Proposition 2 *is true or false if V starts observation in T and receives all responses until $T + \Delta T$.*

Proposition 2 has the same assumption for entity authentication as Proposition 1.

Relay attack-resistance is that a distance verification scheme can resist against a relay attack.

Adaptability is that a distance verification scheme can be adapted to practical systems. Accordingly the scheme must consider a communication processing delay of P (e.g. error correction, modulation, and decoding on communication between V and P) and a secret key processing delay of P for entity authentication.

Modularity is that a distance verification scheme has a framework that is easy to change plural authentication methods.

4. 3. 2. 2. An Attack Model

The study supposes that an only purpose of attacks is to cheat V into accepting a fake location L_i' of N_i. And an attacker is a holder of N_i or a third party. The attacks are classified into the following types.

Attack on entity:

Tampering is an attack that analyzes and forges N or PN directly.

Black box attack is an attack that guesses secret information of N or PN from output of N or PN for various inputs.

Impersonation is an attack that impersonates against N or PN.

Attack on communication:

Message forgery is an attack forges messages on communication channels.

Message replay is an attack replays messages on the channels.

Message delay is an attack delays message on the channel.

Man in the middle attack is an attack that an attacker lies between N_i and PN, and the attacker each impersonates the entities.

Relay attack is an attack that RS_i controlled by N_i relays messages on communication between N_i and PN.

4. 3. 2. 3. Data Structure of a Challenge and a Response

In Section 4.2, for preventing relay attack the authors design the structure of challenge messages, which allows a delay time of transmission to increase if the challenge messages are relayed. They assume that relay delay always occurs if communication is relayed, and the relay delay increases in the following cases:

1. an increase of size of relayed communication data,
2. an increase of the number of relays, and

3. an increase of relay distance between N_i and RS_i.

The study adopts their structure and assumptions in this thesis.

4. 3. 3. The Proposed Schemes

This proposed schemes are based on the location verification scheme [AM05-1] that consists of a framework and plural authentication methods. The proposed schemes extend the framework to verify locations of plural provers, and use the authentication methods intact. Therefore this section does not explain the detail of the authentication methods.

The thesis proposes two schemes and the two variations. The formers are a location verification scheme between two points, which can verify locations of two nodes N_1 and N_2, and a location verification scheme among n points, which can verify locations of n (≥ 2) nodes N_1, …, N_n. Also the latters are a variation that P_i sends a response message to V via P_j (the thesis called "turn type"), and another variation that P_i sends a response message to V directly (the thesis called "loop type").

4. 3. 3. 1. Notation

This subsection defines notations for describing this proposed schemes as follows:

A_Enc(*Y, m*): a secure asymmetric encryption function, which outputs a ciphertext of a message *m* using a public key *Y*. Here, only the corresponding private key *X* can decrypt the ciphertext.

S_Enc(*K, m*): a secure symmetric encryption function, which outputs a ciphertext of a message *m* using a shared key *K*. Here, only *K* can decrypt the ciphertext.

A_Sig(*X, m*) is a secure signature generation function, which outputs a signature of the message *m* using a private key *X*. Here, the validity of the signature is checked using the corresponding public key *Y*.

S_Sig(*K, m*) is a secure MAC (Message Authentication Code) generation function, e.g. HMAC, which outputs a MAC of a message m using a shared key *K*. Here, *K* can only check validity of the MAC.

H(*m*) is a secure one-way hash function, which outputs a hash value of a message *m*.

$\mathbf{Mask_Y}$(*R, Y*) is a public key mask function that masks a public key *Y* using a random number *R*. It is hard to obtain the public key *Y* from a masked public key *Y*' without *R*.

$\mathbf{Mask_X}$(*R, X*) is a secret key mask function that masks a secret key *X* using a random number *R*. It is hard to obtain the secret key *X* form a masked secret key *X*' without *R*.

(Q_i, A_i) is a pair of a question and an answer for N_i.

Req_i is a request message generated by N_i.

Cha_i is a challenge message generated by *PN* for N_i. The message may include a commitment(s) generated by N_i and another node N_j.

Res$_i$ is a response message generated by N_i. The message may include a commitment(s) generated by N_i and another node N_j.

ReqG is a request message generator, which outputs Req_i from an authentication type C. C indicates the type as follows: α is "No authentication", β is asymmetric key-based "Authentication", γ is symmetric key-based "Authentication", and ε is "Authentication with anonymity".

ChaG_*C* is a challenge message generator, which outputs Cha_i from a random number and Req_i. C is an authentication type.

ResG_*C* is a response message generator, which outputs Res_i from Cha_i and a secret key of N_i. C is the same meaning as one of ChaG_C.

VeriF_*C* is a verification function that verifies inputs (a response message and a communication time), and then outputs "*Accept*" if the inputs are valid. The function outputs a value except "*Accept*" if the inputs are invalid.

The above functions and generators are the same as Section 4.2.

VeriF$_{relay}$_*C* is a verification function that verifies inputs (a relayed response message and a communication time including a relay time), and then outputs "*Accept*" if the inputs are valid. The function outputs a value except "*Accept*" if the inputs are invalid.

ComF_*C* is a commitment function that outputs a commitment (signature or MAC) from messages and a secret key of N_i. In short, the function is equal to A_Sig(X, m) or S_Sig(K, m).

t$_{Cha_i}$ is a time that PN sends Cha_i to N_i.

t$_{Cha_i_j}$ is a time that PN sends Cha_j to N_i.

t$_{Res_i}$ is a time that PN receives Res_i form N_i.

t$_{Res_i_j}$ is a time that PN receives Res_j form N_i.

T$_{Relay}$ is a relay delay time of an attacker. $T_{Relay} = 0$ if the attacker does not relay a challenge message or a response message.

T$_{Relay_i}$ is a relay time of N_i, which is approved by PN in advance. Note that the relay time is all processing time for relaying a challenge message and a response message, including a time for committing the messages.

T$_{Delay_i}$ is a response time of N_i, which is approved by PN in advance. Note that the response time is all processing time for generating and sending a response message, including a time for committing the message.

T$_{Com_i}$ is a communication time between N_i and PN.

T$_{Com_i_j}$ is a communication time between N_j and PN via N_i, not including T_{Relay_i} and T_{Delay_j}.

T$_{permit}$ is an error time of verifying locations, which is allowed by PN.

ΔT is a difference time between t_{Res_i} and t_{Cha_i}. Here, $\Delta T = t_{Res\_i} - t_{Cha\_i}$.

s is a speed of RF.

r is an available RF distance of *PN*.

4. 3. 3. 2. A Framework of a Turn Type Location Distance Scheme Between Two Points

This subsection explains a framework in case that a proxy node *PN* verifies locations of nodes N_i and N_j ($i \neq j$). Figure 4.3.1 shows sequences of the proposed schemes.

Step1: N_i generates a request message Req_i using the following equation, and then sends Req_i to *PN*.

$Req_i = \mathrm{ReqG}(Z_i)$

Case 1 "No authentication": $Z_i \leftarrow i \parallel R_i$. Here, "||" indicates concatenation of data, and $B \leftarrow A$ means that A is substituted into B.

Case 2 Symmetric key-based "Authentication": $Z_i \leftarrow i$.

Case 3 Asymmetric key-based "Authentication": $Z_i \leftarrow Y_i$.

Case 4 "Authentication with anonymity": $Z_i \leftarrow Y_i$' (is output of $\mathrm{Mask_Y}(R_i, Y_i)$).

Step2: *PN* receives Req_i, and then obtains Z_i from Req_i. *PN* decides a type of authentication requested by N_i from Z_i. *PN* sends "*Reject*" to N_i if *PN* does not accept the type, and then *PN* stops this protocol.

Step3: *PN* selects a random number R_{PN_i}, and then generates a challenge message Cha_i (includes a question Q_i) and the corresponding answer A_i using the following equation:

$(Cha_i, A_i) = \mathrm{ChaG\_}C(R_{PN\_i}, z_i)$.

$z_i \leftarrow R_{PN}$ if Z_i is equal to $i \parallel R_{PN\_i}$.

$z_i \leftarrow K_i$, *PN* selects the corresponding shared key K_i from i if Z_i is equal to i.

$z_i \leftarrow Y_i$ if Z_i is equal to Y_i.

$z_i \leftarrow Y_i$' if Z_i is equal to Y_i'.

Finally, *PN* sends Cha_i to N_i and then stores a time of sending Cha_i into t_{Cha_i}. Note that *PN* must send Cha_i to N_i in order from MSB (Most Significant Byte).

Step4: N_i receives Cha_i, and then generates a response message Res_i (includes the corresponding answer A_i') using the following equation:

$Res_i = \mathrm{ResG\_}C(Cha_i, \zeta_i)$

$\zeta_i \leftarrow R_{PN\_i}$ if Z_i is equal to $i \parallel R_{PN\_i}$.

$\zeta_i \leftarrow K_i$ if Z_i is equal to i.

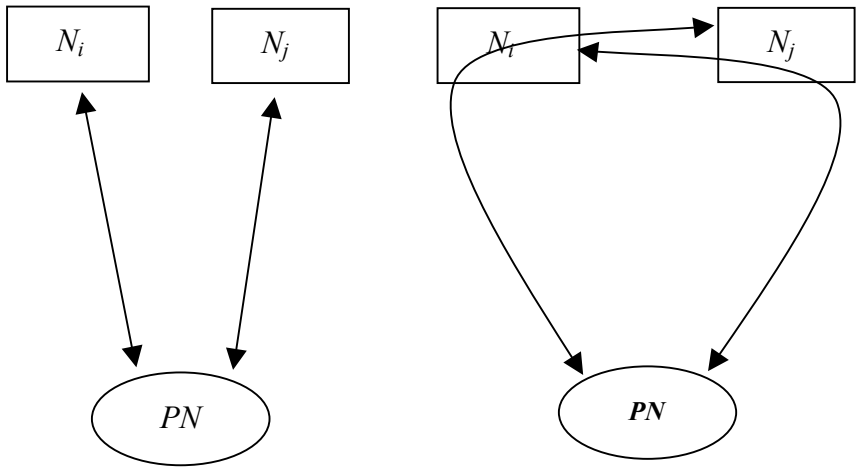
$\zeta_i \leftarrow X_i$ if Z_i is equal to Y_i.

$\zeta_i \leftarrow X_i$' (is output of $\mathrm{Mask_X}(R_{PN\_i}, X_i)$) if Z_i is equal to Y'.

Finally, N_i sends Res_i to *PN*.

Step5: *PN* receives Res_i, and then stores a time of receiving Res_i into t_{Res_i}. Finally *PN* informs ID *j* to N_i.

Step6: *PN* executes a procedure from Step1 to Step5 for N_j and then obtains t_{Cha_j} and t_{Res_j}. Finally *PN* informs Z_i to N_j. The above protocol is the same with the scheme [AM05-1]. Figure 4.3.1. a) shows procedures from Step1 to Step5 for N_i and N_j.



a) Step1 – Step 5 b) Step7 – Step11

Fig. 4.3.1. Sequences of a turn type distance verification scheme between two points.

Step7: *PN* selects a random number $R_{PN\_j\_i}$, and then generates a challenge message Cha_{j_i} and the corresponding answer A_i using the following equation:

$(Cha_{j\_i}, A_i) = \text{ChaG\_}C(R_{PN\_j\_i}, z_i)$.

Here, z_i is equal to one of Step3. Next, *PN* sends Cha_{j_i} to N_j and then stores a time of sending Cha_{j_i} into $t_{Cha\_j\_i}$.

Step8: N_j receives Cha_{j_i}, and then generates a commitment Cmt_{j1} using the following equation:

$Cmt_{j1} = \text{ComF\_}C(Cha_{j\_i}, \zeta_i)$.

Here, ζ_i is equal to one of Step4. Next, *PN* sends $Cha_{j\_i} \| Cmt_{j1}$ to N_i.

Step9: N_i receives $Cha_{j\_i} \| Cmt_{j1}$, and then generate a response message Res_{j_i} and a commitment Cmt_{i1} using the following equations:

$Res_{j\_i} = \text{ResG\_}C(Cha_{j\_i}, \zeta_i)$, and
$Cmt_{i1} = \text{ComF\_}C(Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1}, \zeta_i)$.

Next N_i sends $Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1} \| Cmt_{i1}$ to N_j.

Step10: N_j receives $Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1} \| Cmt_{i1}$, and then generates a commitment $Comt_{j2}$ using the following equation:

$Cmt_{j2} = \mathrm{ComF}\_C(Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1} \| Cmt_{i1}, \zeta_i)$.

Next N_j sends $Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1} \| Cmt_{i1} \| Cmt_{j2}$ to PN.

Step11: PN receives $Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1} \| Cmt_{i1} \| Cmt_{j2}$, and then stores a time of receiving the message into $t_{Res\_j\_i}$.

Step12: PN executes a procedure from Step7 to Step11 for N_j and then obtains $t_{Cha\_i\_j}$ and $t_{Res\_i\_j}$. Figure 4.3.1. b) shows procedures from Step7 to Step11 for N_i and N_j.

Step13: PN calculates ΔT and T_{Com} as follows:

$$\Delta T_{\mathrm{i}} = t_{Res\_i} - t_{Cha\_i}, \tag{4.3.1}$$

$$\Delta T_{\mathrm{j}} = t_{Res\_j} - t_{Cha\_j}, \tag{4.3.2}$$

$$T_{Com\_j\_i} = t_{Res\_j\_i} - t_{Cha\_j\_i} - 2 \cdot T_{Relay\_j} - T_{Delay\_j}, \tag{4.3.3}$$

$$T_{Com\_i\_j} = t_{Res\_i\_j} - t_{Cha\_i\_j} - 2 \cdot T_{Relay\_i} - T_{Delay\_i}. \tag{4.3.4}$$

Finally, PN can obtain distances d_i, d_j, and $d_{\{i,j\}}$ using the following equations (4.3.10)(4.3.11)(4.3.12) if all the following conditions (4.3.5)(4.3.6)(4.3.7)(4.3.8)(4.3.9) are satisfied.

$$| T_{Com\_j\_i} - T_{Com\_i\_j} | \leq T_{Permit}, \tag{4.3.5}$$

$$\text{“}Accept\text{”} = \mathrm{VeriF}\_C(A_i, Res_i, z_i, r, s, T_{Delay\_i}, \Delta T_i), \tag{4.3.6}$$

$$\text{“}Accept\text{”} = \mathrm{VeriF}\_C(A_j, Res_j, z_j, r, s, T_{Delay\_j}, \Delta T_j), \tag{4.3.7}$$

$$\text{“}Accept\text{”} = \mathrm{VeriF}_{\mathrm{relay}}\_C(A_i, Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1} \| Cmt_{i1} \| Cmt_{j2}, z_i, z_j, r, s, T_{Com\_j\_i}), \tag{4.3.8}$$

$$\text{“Accept”} = \mathrm{VeriF}_{\mathrm{relay}}\_C(A_j, Res_{i\_j} \| Cha_{i\_j} \| Cmt_{i1} \| Cmt_{j1} \| Cmt_{i2}, z_j, z_i, r, s, T_{Com\_i\_j}), \tag{4.3.9}$$

$$d_i \ (= (\Delta T_i - T_{Delay\_i}) / 2s), \tag{4.3.10}$$

$$d_j \ (= (\Delta T_j - T_{Delay\_j}) / 2s), \tag{4.3.11}$$

$$d_{\{i,j\}} \ (= ((T_{Com\_j\_i} + T_{Come\_i\_j}) / 2s - d_i - d_j) / 2). \tag{4.3.12}$$

4. 3. 3. 3. A Framework of a Loop Type Distance Verification Scheme Between Two Points

A loop type scheme differs from a turn type scheme in the following steps: Step9 is that N_i sends PN to $Res_{j\_i} \| Cha_{j\_i} \| Cmt_{j1} \| Cmt_{i1}$ directly, and Step10 is ignored (Figure 4.3.2 shows a sequence of the loop type scheme). Therefore, PN should calculate $T_{Com\_j\_i}$ and $T_{Com\_j\_i}$ as follows:

$T_{Com\_j\_i} = (t_{Res\_j\_i} - t_{Cha\_j\_i}) - (T_{Relay\_j} - T_{Delay\_i})$,

$T_{Com\_i\_j} = (t_{Res\_i\_j} - t_{Cha\_i\_j}) - (T_{Relay\_i} - T_{Delay\_j})$.

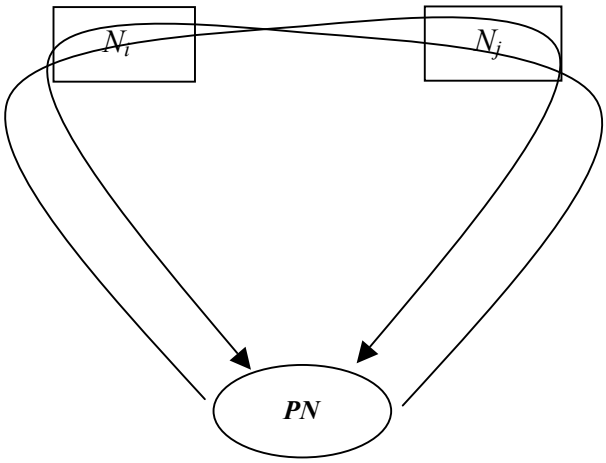


Fig. 4.3.2. A sequence of a loop type distance verification scheme between two points.

4. 3. 3. 4. A Framework of Distance Verification Schemes Among n Points

A distance verification scheme among n points is that it executes a distance verification scheme between two points for each combination W (see subsection 4.3.2.1).

4. 3. 3. 5. Requirements of a Function and Generators

This subsection describes a detail of a function $\text{VeriF}_{\text{relay}}\_C$, however the study omits details of other functions and generators are described in Section 4.2.

$\text{VeriF}_{\text{relay}}\_C$ can output "*Accept*" if inputs (A_i, $Cha_{j\_i} \parallel Res_{j\_i} \parallel Cmt_{j1} \parallel Cmt_{i1} \parallel Comt_{i2}$, z_j, z_i, r, s, $T_{Com\_j\_i}$) are valid, otherwise the function outputs a value except "*Accept*". "Valid" means the inputs satisfy the following conditions:

$2r \geq d_{\{i,j\}}$ ($= T_{Com\_j\_i} / 2s$) > 0,

an answer A_i (or Signed A_i) is valid (how to verify the answer is the same with a verification function VeriF_C of the scheme described in Section 4.2,

a result of verifying a commitment Cmt_{j1} (signature or MAC) using a plain text Cha_{j_i} and z_i is valid,

Cha_{j_i} is equal to one generated by *PN*,

a result of verifying a commitment Cmt_{i1} using a plain text $Res_{j\_i} \parallel Cha_{j\_i} \parallel Comt_{j1}$ and z_i is valid, and

a result of verifying a commitment Cmt_{j2} using a plain text $Res_{j\_i} \parallel Cha_{j\_i} \parallel Comt_{j1} \parallel Comt_{i1}$ and z_i is valid.

4. 3. 3. 6. Viability of Our Proposed Schemes

These proposed schemes satisfy our all requirements.

Distance verifiability between two points and Distance verifiability among n points: the thesis explains security of this requirement on Section 4.3.4. This section considers the realization of out proposed scheme. The proposed scheme requires a precision of 3.333… [nsec] per 1 [m], and the approved processing delay is a maximum of 0.333… [nsec] if an error range is within 10 [cm]. In the paper [MK03], a PC with improved software has a time precision of 1 [μsec], and a PC with improved hardware (e.g. a high-frequency crystal oscillator) has a time precision of 34 [ns]. In short, the PC with improved hardware can measure a distance by about 10 [m]. Moreover, the paper [TMIK05] proposed a special time-stamp hardware has a time precision of 8 [ns] and therefore a PC with the hardware can measure a distance by 2.4 [m]. On the other hand, the proposed scheme is a kind of TOA and TDOA. Here, the paper [OTWFYSK03] proposed a system that obtains location by 1-4 [m] of a wireless LAN node using TDOA and plural access points (i.e. verifiers). Moreover, the papers [KNF03][HMYMMA04] show to be able to simulate an influence of multipass fading. Therefore the study thinks to be able to realize the proposed scheme that measures a distance by 1-10 [m].

Relay attack-resistance: the thesis explains on Section 4.3.4.

Adoptability: these proposed schemes assume a proxy node knows processing and relay delays of nodes, and the proxy node considers the delays when the proxy node verifies locations of the nodes. Thus the proposed schemes satisfy “Adoptability”.

Modularity: the proposed schemes are that a node can easily select one of six authentication methods. Therefore, the proposed schemes satisfy “Modularity”.

4. 3. 4. Security Analysis

On security of distances d_i and d_j the proposed schemes are equal to the scheme described in Section 4.2. Thus the study discusses a distance $d_{\{i, j\}}$ between N_i and N_j. The proposed schemes satisfy “Distance verifiability between two points and distance verifiability among *n* points” and “Relay attack-resistance”, based on the following assumptions:

Assumption 1: *PN* knows relay times (T_{Relay_i}, T_{Relay_j}) and delay times (T_{Delay_i}, T_{Delay_j}). The delay time consists of a secret key processing time that N_i needs to calculate A_i (or Signed A_i) and a communication processing time of N_i. Attackers are hard to change T_{Relay_i}, T_{Relay_j}, T_{Delay_i} or T_{Delay_j}. Note that the proposed schemes require tamper-resistance to N_i for satisfying this assumption.

Assumption 2: Attackers cannot transmit data faster than s. s is equal to a speed of light because of using RF.

Assumption 3: A relay delay (its time is T_{Relay} (> 0)) occurs if communication between (*PN* and N_i), (*PN* and N_j), or (N_i and N_j) is relayed, and T_{Relay} satisfies an inequality ((((T_{Relay} + $T_{Com\_j\_i}$ +

$T_{Com\_i\_j}$) / $2s - d_i - d_j$) / 2) > $2r$. The scheme [AM05-1] has challenge and response messages that can force relay delay to increase when communication is relayed. The proposed schemes satisfy this assumption since the proposed schemes are based on the scheme described in Section 4.2. Note that $T_{Com\_j\_i}$ (or $T_{Com\_i\_j}$) includes T_{relay_i} (or T_{relay_j}) supposed by *PN*.

Assumption 4: RS_i and third parties cannot know a secret key of N_i. Note that the study assumes A_Enc(Y, m), S_Enc(K, m), A_Sig(X, m), S_Sig(K, m), and H(m) are secure, and N_i is a tamper-resistant module for satisfying this assumption.

Assumption 5: A least party of N_i and N_j is honest.

This proposed schemes decide that a distance between N_i and N_j is within $d_{\{i,j\}}$ if conditions (4.3.5)(4.3.6)(4.3.7)(4.3.8)(4.3.9) are satisfied. Using (4.3.3)(4.3.4)(4.3.5)(4.3.12) $d_{\{i,j\}}$ is expanded as follows:

$$d_{\{i,j\}} = ((T_{Com\_j\_i} + T_{Com\_i\_j}) / 2s - d_i - d_j) / 2$$

$$= ((t_{Res\_j\_i} - t_{Cha\_j\_i} - 2T_{Relay\_j} - T_{Delay\_j} + t_{Res\_i\_j} - t_{Cha\_i\_j} - 2T_{Relay\_i} - T_{Delay\_i}) / 2s - d_i - d_j) / 2,$$

where $2r \geq d_{\{i,j\}} > 0$, and $| T_{Com\_j\_i} - T_{Com\_i\_j} | \leq T_{Permit}$.

The section explains all expanded parameters. From Assumption 1 attackers are hard to change T_{Delay_i}, T_{Delay_j}, T_{Relay_i}, and T_{Relay_j}. From Assumption 2 attackers cannot decrease $T_{Com\_j\_i}$ and $T_{Com\_i\_j}$. From Assumption 3 T_{Relay} satisfies (((T_{Relay} + $T_{Com\_j\_i}$ + $T_{Com\_i\_j}$) / $2s - d_i - d_j$) / 2) > $2r$ if communications are relayed. From Assumption 5 the inequality $| T_{Com\_j\_i} - T_{Com\_i\_j} | \leq T_{Permit}$ is not satisfied if N_i or N_j delays own response message so that d_i or d_j increases. In other words, if the proposed schemes ignore Assumption 5 or the inequality (4.3.5), a dishonest node can decrease $d_{\{i,j\}}$ to delay own response message.

On the other hand, VeriF_{relay}_C is secure because only N_i can calculate A_i (or Singed A_i) from Assumption 4.

Consequently, this proposed schemes can decide that a distance between N_i and N_j is within $d_{\{i,j\}}$ if conditions (4.3.5)(4.3.6)(4.3.7)(4.3.8)(4.3.9) are satisfied.

4. 3. 5. A Variety of Location Verification Schemes

4. 3. 5. 1. How to Force a Verification of N_j to Depend on One of N_i

To innovate on security of the proposed schemes, the study considers connecting verifications of N_i and N_j as follows:

1. *A proxy node PN* can generate a challenge message of Step 6 (or Step 12) on Section 4.3.3.2 to depend on a response message of Step 5 (or Step11), or
2. *PN* can design a response message of Step 6 (or Step12), as N_i (or N_j) cannot generate the

response message without challenge and response messages from Step1 to Step5 (or from Step 7 to Step11).

The study supposes that the above-mentioned ways guarantee the continuity between verifications.

4. 3. 5. 2. How to Decrease an Amount of Transmission Using Broadcast

PN can decrease an amount of transmission broadcasting a challenge message to plural nodes. However *PN* may simultaneously receive the corresponding response messages from the nodes. Thus *PN* should have ability of plural receiving and processing.

On the proposed location verification scheme among n points, plural nodes can decrease an amount of transmission broadcasting relayed challenge and response messages other nodes and *PN*. Similarly the nodes should have ability of plural receiving and processing.

4. 3. 5. 3. How to Realize "Sheltered Location"

Equally the scheme described in Section 4.2, *PN* can issue a location certificate that consists of distances and IDs of N_i and N_j, a time to obtain the information, ID (or location information) of *PN*, and the corresponding signature generated by *PN* on the proposed schemes. If *PN* does not include the ID (or the location information) of *PN* in the location certificate, the certificate can only prove d_i, d_j and $d_{\{i, j\}}$ at the time, moreover a third party cannot trace a location of N_i and N_j from the certificate. The thesis calls such property "Sheltered location".

4. 3. 5. 4. How to Expand a Verifiable Range Using Plural Proxy Nodes

If the proposed schemes assume plural *PN*s that can communicate mutually, the schemes can expand own verifiable range and realize "Region verification, Position verification, and Route verification" like the scheme described in Section 4.2. For example, a node 1 relays a challenge message that PN_A sends to a node 2, and then the node 2 sends the corresponding response message to PN_B, finally PN_A and PN_B share the sending time, the receiving time, the messages, and so on.

4. 3. 5. 5. How to Share a Relayed Node on a Distance Verification Scheme Among n Points

On a distance verification scheme among n points, if *PN* decides the only one relayed node N_i from all nodes, *PN* can prove that all the nodes except the relayed node exist within a distance from the relayed node. However, N_i can easily force *PN* to accept a fake distance that is farther than a real distance if N_i is dishonest. Therefore, *PN* should use this way in case of that N_i is honest.

4. 3. 5. 6. How to use plural relayed nodes

Section 4.3.3 supposes a relayed node per one pair of challenge and response messages. The study

expects that *PN* can obtain more detailed location information among nodes if more than two relayed nodes relay the pair. On the other hand, this way may force verification precision of these proposed schemes to deteriorate because of the complicated procedures.

4. 3. 6. Conclusion

This thesis proposed plural provers verifiable location verification schemes to expand the location verification scheme described in Section 4.2. The proposed scheme can prove location relation between plural provers since a prover relays a challenge message for another prover.

Chapter 5. Secure Cryptographic Location-Based Services

5. 1. Cryptographic Location-Based Services

This section defines cryptographic location-based services (cryptographic LBSs) that utilize cryptography and shows instances of cryptographic LBSs.

Recently, services for utilizing real context of mobile nodes have been studied actively on ubiquitous computing / networks. Especially, the study expects the services, which use location information of nodes as real context, to come into wide use in the future. Such services are called location-based services (LBSs). Services of LBSs include information distribution to a specific location, navigation of walkers, tracking of mobile nodes [ITWUSM00][WTTUM96], location-based access control [CN02], and issuing of location certificates [WF03], along with other applications.

On the other hand, according to the wide use of LBSs, many LBSs (e.g. high-value LBSs) require more security; for that reason, LBSs require cryptographic capability. The section shows instances of cryptographic LBSs as follows:

a system that a user can read a secret business document stored in a notebook PC in an office however cannot read the document when the notebook PC goes out of the office because of re-encrypting the document,

a system that a user can use a digital sign key (as an official seal of a company) stored in a device in an only specific room, and

a system that issues a location certificate including a system signature, a meeting location, a meeting time, meeting participants names and a group signature, which all the participants of the meeting generate, for proving a fact the participants join in the meeting to a third party.

Cryptographic LBSs require key management (e.g. key sharing for session encryption between a provider and a node) and location management (e.g. location verification of a node), and cryptographic LBSs compose mainly of a key management function and a location management function. As a result, cooperation between the key management and location management functions realizes cryptographic LBSs. However these functions have mostly been studied individually. This study indicates that cryptographic LBSs are insecure in due to incomplete integration of key

management and location management functions, for example a user would like to share a key with a mobile node of a specific location however an attacker of other location forces the user to share the key with the attacker. Therefore the study considers security of integration between key management and location management for realizing secure cryptographic LBSs.

The secure integration demands the following requirements:

key management and location management functions can authenticate the same node,

key management and location management functions are indivisible, and

key management and location management functions must be controlled by valid policy.

5. 2. How to Construct Secure Cryptographic Location-Based Services

5. 2. 1. Introduction

5. 2. 1. 1. Background

Recently, services for using the real context of mobile nodes have been studied actively on ubiquitous computing / networks. The study expects such services, which use location information of nodes as a real context, to come into wider use in the future. Such services are called location-based services (LBSs). LBSs include information distribution to a specific location, navigation of walkers, tracking of mobile nodes [ITWUSM00][WTTUM96], location-based access control [CN02], and issuing of location certificates [WF03], along with other applications. Many high-value LBSs require security; for that reason, LBSs require cryptographic capability. As instances of cryptographic LBSs, the study postulates a system in which a user can read a secret business document stored in a notebook PC in an office. However, in this system, re-encryption of the document prevents its reading when the notebook PC is removed from the office. With this system, a user can use a digital sign key (as an official company seal) stored in a device for a specific room. Such cryptographic LBSs comprise a key management function (e.g. key sharing for session encryption between a provider and a node) and a location management function (e.g. location verification of a node). Key management methods have been studied variously until now. As location management location measurement technologies using global positioning system (GPS) and radar are realized, technologies that use a wireless LAN and radio frequency identification (RFID) are advancing apace. However, these key and location management methods have mostly been studied individually. Therefore we consider security of integration between key management and location management for realizing secure cryptographic LBSs.

5. 2. 1. 2. Location Management

Location measurement technologies have made the transition from methods [GW98] using GPS, a base station (of cellular phone systems) and radar for outdoors to methods [KNF02][NNT03][VN04] using a wireless LAN, RFID and sensor networks for indoors. In addition, papers [AM05-1] [AM05-2][BC93][CH04][SSW03][WF03] proposed secure location verification schemes using communication delay. These technologies differ in available ranges, costs and security. For instance,

a GPS satellite is expensive, but an RFID tag inexpensive.

This thesis refers to digitized location information as a “Location Token”. The study assumes a location token model, which comprises plural location token providers (with various location measurement functions), provers (which prove their own location using location tokens) and verifiers (which verify locations of provers using location tokens). For example, one study [WF03] adopts a narrowly-defined location token mode that supposes one location measurement scheme. This paper uses location management as a generic term to refer to the following functions:

- Location verification: a verifier directly verifies a mobile node location in real-time, and
- Location certification: mobile nodes prove their own locations to a third party using location tokens.

The study presumes a model that consists of servers (e.g. proxy nodes), provers (e.g. mobile nodes), and verifiers. The server provides a location data (i.e. location token) of the prover to the prover using various location measurement techniques, the prover prove own location to the verifier using the location token, and the verifier verify the location of the prover. On such a model the verifier may not verify the location token because the verifier may not have all the corresponding verification methods. The paper [AM05-7] proposed a location certification infrastructure that distributes location tokens and solves this problem.

5. 2. 1. 3. Key Management

In this paper, key-management targets are the following keys:

- A node private key is a unique secret key of a mobile node (and the corresponding public key); the mobile node securely stores the secret key, and
- A processed key: is the output of a key-management function that is an inputted node private key dependent on information, the output is a secret key (and the corresponding public key).

In this paper, key management is a generic term used to refer to the following functions:

- Key issuing: is a method that issues a key, e.g., issuance of a public key certificate from a Certificate Authority (CA);
- Key sharing: is a method by which plural entities share the same key, for instance Diffie-Hellman key exchange scheme (“key sharing” includes “key exchange”);
- Key distribution: is a method that distributes a key to specific entities, for instance broadcast encryption;
- Key generation: is a method in which one or more entities generates a key, for instance RSA key generation;
- Key revocation: is a method that revokes a key, for instance broadcast exclusion and Certificate Revocation List (CRL); and

 Key control: is a method that controls access to a key, for instance Kerberos.

A term “key operating” means it executes each function above.

5. 2. 1. 4. Integration of Key Management and Location Management

A few studies examine a mix of key management and location management functions. One study [CN02] proposed a PC system, in which a PC hard disk is decrypted because a personal radio device allows the PC to use a decryption key if their authentication is successful, when the device closes in the PC. Another study [BM02] proposed trusted access points measuring the location of a mobile node and shares a key using electric field intensity of beacons sent by the access points. These studies are schemes that mix location management and key management functions. However the studies do not clarify the structure of integration between the two functions. Therefore, the study cannot analyze structural security. Incomplete integration might cause an attack on cryptographic LBSs, for instance a provider would like to share a key with a mobile node on a specific location. However, an attacker on other location may force the provider to share the key with the attacker after the valid node location verification. Therefore, the study considers security of integration between key management and location management for realizing secure cryptographic LBSs. Here, this paper treats the following problems:

1) An attacker may impersonate a valid node if target nodes are not the same on key management and location management functions,
2) An attacker may replace a valid function with an invalid function if key management and location management functions are indivisible; and
3) A provider may not provide valid LBSs if a function execute after execution of another function is a failure.

5. 2. 1. 5. Our Goal

The cryptographic LBSs require key management and location management functions. However secure integration of these functions has not been clarified because the functions have only been studied individually until now. Therefore, this paper proposes a method of integrating key management and location management functions for realizing secure cryptographic LBSs. In addition, the study suggests new cryptographic LBSs by assessing all combinations of key management and location management.

This proposed method defines two general integrated functions (location key function). A first function is a key management function that is executed by depending on output of a location management function. A second function is a location management function that is executed by depending on output of a key management function. Our assumed system consists of a location key proxy node that provides services using location key functions and a node that requests a service to

the proxy node. For realizing secure cryptographic LBSs, the study applies the following approaches:

1) agreement of target nodes by a context connection (CC) value;
2) improvement of mutual dependence by a construct that inputs output of a management function to another management function; and
3) policy-based access control for functions.

5. 5. 2. Requirements

5. 5. 2.1. Location Key

In this paper, a location key function means an integrated function of key management and location management functions. The location key function is classified as follows:

- **LK** (**L**ocation operation, then **K**ey operation) function – is a key management function that is inputted to output of a location management function; and
- **KL** (**K**ey operation, then **L**ocation operation) function - is a location management function that is inputted to output of a key management function.

LK and KL functions output a pair of a location key and its corresponding location token. Note that an LK function does not output a location token basically. A location key is a processed key: that is outputted from an LK function; or is targeted by a location token (is outputted from a KL function). The location keys are determined from location between a node location, a time when location key function is executed, and a node private key.

5. 5. 2. 2. Entities

This proposed method consists of the following entities:

- **Node**: is a mobile node that requests services using location key functions to a proxy node. A node stores a unique node private key securely and his ID is "*i*". In addition, the node might obtain his own location information, time information, and random numbers. This study supposes a cellular phone, PDA and a notebook PC as nodes.
- **Proxy node**: is a proxy node that provides services using location key functions to a node. A proxy node has a KL function and an LK function as location key functions, and stores a unique proxy node private key securely; the ID is "*j*". The proxy node provides services using location key functions according to a location key policy to a node. In addition, a proxy node might obtain its own location information, time information, and random numbers. This study supposes the following as proxy nodes:

  1. **Station**: is a trusted apparatus that has high performance and is fixed on a specific location.

This study presumes a base station of cellular phone systems and an access point (or a PC that connects to an access point) of a wireless LAN as a station.

2. **Mobile**: is a mobile node that has middle performance. This study supposes a notebook PC, a PDA, or a cellular phone as a mobile. A node does not necessarily trust a mobile.
3. **Sensor**: is a fixed node that has low performance and is active. This study supposes a sensor node of sensor networks and an active IC tag (e.g. a smart tag) of RFID as a sensor. A sensor is not highly trusted by a node.
4. **Tag**: is a device that has little performance and is passive. This study supposes a tag of RFID as a tag. A node does not completely trust. In case of wearing a tag, a reader / writer writes a location token to the tag. Therefore, the tag is a node and the reader/writer is a proxy node.

5. 2. 2. 3. Location Measurement Techniques

This study classifies methods that measure locations of nodes into the following types:

- **Report type**: means that a method in which a node reports self-obtained location of the node to a proxy node. This study presumes a node supporting system, for example GPS.
- **Inference type**: means that a method in which a proxy node infers location of a node from evidence (e.g. IDs of tags). This study generally presumes methods using RFID.
- **Direct type**: means that a method in which a proxy node directly verifies location of a node in real-time. This study presumes methods using radar, a wireless LAN, and communication delay.

5. 2. 2. 4. Location Token

A location token is digitized location information obtained from a proxy node in Section 5.2.2.2 and location measurement technique in Section 5.2.2.3. The location token includes ID of a node, location information of a node and time information when a location key function executes. A location token also includes a location key, or a location key function issues a pair of a location token and the corresponding location key. This study assumes the following location tokens:

- **A location certificate**: is a kind of public key certificate with which a station or a mobile signs IDs of a node and a proxy node; location information of a node, a location key and time information when a location key function executes. Apparently, that a location certificate is a kind of time-stamp [HS90] that includes location information and a location key. The location certificate supposes direct type location measurement technique.
- **Location evidence**: is digitized location information with a message authentication code (MAC) that a node obtains from a sensor. Use of location evidence supposes inference-type location measurement technique. Therefore the location evidence includes an ID of sensor or location information of a node. In addition, the location evidence might include a node ID, a location key

and time information when a location key function executes.

- **Provisional location evidence**: is digitized location information that a node obtains from a tag. The provisional location evidence supposes inference-type location measurement technique. Therefore, provisional location evidence includes an ID of a tag.
- **A location reference**: is digitized location information that a node self-calculates using supporting entities (e.g. GPS). The location reference supposes report-type location measurement technique. Note that a proxy node issues no location reference and the proxy node might transform the received location reference to other type location token.

Table 5.2.1 show the relation between location tokens, proxy nodes and location measurement techniques. A hyphen means that the corresponding technique is nonexistent now. Define the corresponding new location token if a technique that corresponds to the hyphen appears in future. In addition, a station, a mobile and a sensor receive a location reference and can then transform the location reference to a location certificate or location evidence.

Table 5.2.1 Relation between location tokens, proxy nodes and location measurement techniques

Techniques / Proxy nodes	Report type measurement	Inference type measurement	Direct type measurement
Station	Location reference	-	Location certificate
Mobile	Location reference	-	Location certificate
Sensor	Location reference	Location evidence	-
Tag	-	Provisional location evidence	-

5. 2. 2. 5. Security assumptions

This paper makes the following security assumptions:

1. A proxy node becomes secure, a station, a mobile, a sensor and a tag in that order. Especially the station is a trusted party.
2. Each key management method and each location management method is secure. The proposed method uses existing key management and location management methods.
3. An attacker is a node or a third party.
4. For attacking, a node and a third party might conspire.
5. A communication channel is not secure: anyone can obtain data on the channel.
6. An attacker purposes location key functions of a proxy node to use illegally, and purposes outputs of the location key functions to use change illegally.

5. 2. 2. 6. Requirement

This thesis designs the proposed method for satisfying the following requirements:

1. ***Availability***: is that only allowed nodes can use location key functions of a proxy node according to a location key policy.
2. ***Universality***: is that the proposed method is easily adaptable to existing systems. Actual systems, which include various proxy nodes, various location measurement technologies, various key management methods and various location management methods, require the proposed method for universal design.
3. ***Associativity***: is that association between a key management function and a location management function is secure. Consequently, our proposed method solves three problems shown in section 1.4.
4. ***Privacy***: is that information (ID of a node, location information of a node and time information when a location key function executes), which is demanded by a node demands to conceal, is not leaked from a location token. *Privacy* excludes inconsistent requests of a node from consideration. This paper respectively refers to concealing an ID, concealing location information of a node, and concealing time information when a location key function executes "anonymity, location-hiding, and time-hiding" respectively.

5. 2. 3. The Proposed Method

5. 2. 3. 1. Notation

The study next shows the notation for explaining the proposed method:

- ***NK***: is a node private key.
- ***PNK***: is a proxy node private key.
- ***NID***: is an ID of a node.
- ***PID***: is an ID of a proxy node.
- $\boldsymbol{Info_N}$: is node information that consisting of random numbers, location information and time information.
- $\boldsymbol{Info_{PN}}$: is proxy node information that comprising of random numbers, location information, and time information.
- $\boldsymbol{Data_{NK}}$: is data that depends on *NK* (and $Info_N$).
- ***PK***: is a processed key that is the output of a key management function with input of $Data_{NK}$.
- ***LK***: is a location key.
- ***R***: is a random number.
- ***LT***: is a location token.

- ***LKP***: is a location key policy that includes conditions for executing location key functions.
- ***LKS***: is an internal status of a proxy node: the status (e.g. existence of a specific key, the balance and current location of a proxy node) is needed for judging *LKP*.
- ***KMT***: is a type of key management (key issuing, key generation, key sharing, key distribution, key revocation, key control and none).
- ***LMT***: is a type of location management (location verification, location certification and none).
- ***LKT***: is a type of location key management (*KMT* || *LMT* and *LMT* || *KMT*).
- ***LTT***: is a type of location token (a location certificate, location evidence, a location reference, and *none*)
- ***NPT***: is a type of requested node privacy (anonymity, location-hiding, time-hiding, and *none*).
- ***CC*** value: is a context connection value (*NID*, *R*, *PK* or *LT*)

- **KM** function: is a key management function that outputs *PK* for input ($Data_{NK}$, *KMT*, *NID*, *PID*, *PNK*, $Info_N$, and $Info_{PN}$). According to *KMT*, {*NID*, *PID*, *PNK*, $Info_N$ and $Info_{PN}$} cannot be ignored. As *PK*, a KM function outputs *NID* (in case that *KMT* indicates a key management method with node authentication) or *R* (in case that $Info_N$ or $Info_{PN}$ include *R*).
- **LM** function: is a location management function that outputs *LT* for input ($Data_{NK}$, *LMT*, *PID*, *PNK*, $Info_N$, $Info_{PN}$, and *LTT*). According to *LMT*, {*NID*, *PID*, *PNK*, $Info_N$ and $Info_{PN}$} cannot be ignored. The LM function outputs a pre-selected *LT* if *LTT* is *none*. As *LT*, the LM function outputs *NID* (in case that *LMT* indicates a key management method with node authentication) or *R* (in case that $Info_N$ or $Info_{PN}$ include *R*).
- **PJ** function: is a policy judgment function that outputs *KMT*, a pair of {*NPT*, *LMT*, *LTT*} or *Reject* for input (*LKP*, *NPT*, *LTT*, *LKS*, *LKT*, *NID*, and $Info_N$) According to *LKP*, {*NPT*, *LTT*, *LKS*, *LKT*, *NID*, and $Info_N$} cannot be ignored.
- **KL** function: is a location key function that outputs a pair of {*LK*, *LT*} or *Reject* for input ($Data_{NK}$, *LKP*, *LKS*, *LKT*, *NID*, *PID*, *PNK*, $Info_N$, $Info_{PN}$, *LTT*, and *NPT*).The *LT* includes *LK* if *LTT* = location certificate. According to *LKT* and *LKP*, {*LKS*, *LKT*, *NID*, *PID*, *PNK*, $Info_N$, $Info_{PN}$, *LTT*, and *NPT*} cannot be ignored.
- **LK** function: is a location key function that outputs *LK* or *Reject* for input ($Data_{NK}$, *LKP*, *LKS*, *LKT*, *NID*, *PID*, *PNK*, $Info_N$, and $Info_{PN}$). According to *LKT* and *LKP*, {*LKS*, *LKT*, *NID*, *PID*, *PNK*, $Info_N$ and $Info_{PN}$} cannot be ignored.

5. 2. 3. 2. Constructions of a KL function and an LK function

Figure 5.2.1 shows constructions of a KL function and an LK function. The KL function and LK function comprise a KM function, an LM function, and a PJ function.

The KL function consists of a management path that inputs output of a KM function into an LM

function and a control path that inputs output of a PJ function into a KM function and an LM function.

On the management path, a KM function executes a challenge-Response key operation, which is requested by a key management type *KMT*, with a node for input (a node ID *NID*, a proxy node private key *PNK*, proxy node information $Info_{PN}$, a proxy node ID *PID*, and node information $Info_N$); and then the KM function outputs a processed key *PK*. The KM function outputs *NID* (if the KM function performs node authentication) or a random number *R* (if the KM function performs node distinguishing with a temporal ID (i.e. *R*)) as a *CC* value. The *PK* is a *CC* value if the KM function does not output *NID* or *R*. Next, an LM function executes a *CC* value-dependent challenge-Response location operation, which is requested by a location management type *LMT*, with a node for the output of the KM function, and then the LM function outputs a pair of {a location key *LK*, a location token *LT*} that is requested by a node privacy type *NPT* and a location token type *LTT*. Here, a term "*CC* value-dependent" means that the LM function authenticates that the node has *NID*, *R* or the corresponding secret information.

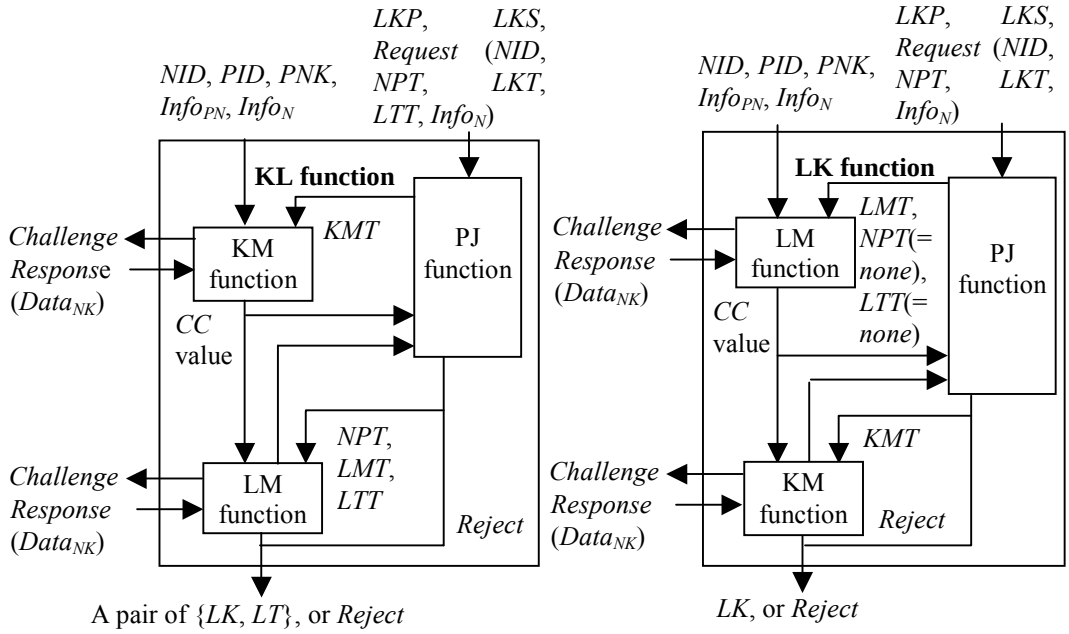


Fig. 5.2.1. Constructions of a KL function and an LK function

On the control path, a PJ function controls execution of a KM function according to a location key policy *LKP*, a location key status *LKS*, and Request of a node. Next, the PJ function controls execution of an LM function according to the output of the KM function, *LKP*, *LKS*, and the Request. If the PJ function outputs *Reject*, the KL function stops execution and outputs *Reject*.

The LK function consists of a management path that inputs output of an LM function into a KM function and a control path that inputs output of a PJ function into an LM function and a KM function.

On the management path, an LM function executes a challenge-response location operation, which is requested by *LMT*, with a node for input (*NID*, *PNK*, $Info_{PN}$, *PID*, and $Info_{N}$); and then the LM function outputs a pre-selected *LT*. The LM function outputs *NID* (if the LM function performs node authentication) or *R* (if the LM function performs node distinguishing with a temporal ID (i.e. *R*)) as a *CC* value. The *LT* is a *CC* value if the LM function does not output *NID* or *R*. Next, a KM function executes a *CC* value-dependent challenge-response key operation, which is requested by *KMT*, with a node for the output of the LM function, and then the KM function outputs *LK*.

On the control path, a PJ function controls execution of an LM function according to *LKP*, *LKS* and the Request of a node. Here, the PJ function inputs *none* as *NPT* and *LTT*. Next, the PJ function controls execution of a KM function according to the output of the LM function, *LKP*, *LKS* and the Request. If the PJ function outputs *Reject*, the LK function stops execution and outputs *Reject*.

Table 5.2.2 shows an example of a location key policy *LKP*. The *LKP* comprises plural records. Each record includes three items: an attribute that is an identification of a node or a belonging of a node, an action that is an allowed operating of location key functions and a condition that is a requirement to allow the action.

Table 5.2.2. An example of a location key policy

Item / Record	Attribute	Action	Condition
Record 1	Company *A*	*KMT* || *LMT*	Location *X*
Record 2	Entity *B*	Any	None
Record 3	Group *C*	*LMT* || *KMT*	¥10,000

5. 2. 3. 3. Sequence of Our Proposed Method

This section explains a sequence of the proposed method (see Figure 5.2.2).

1. A node *i* sends a Request which is *NID* || *LKT* (=*LMT* || *KMT*) || $Info_{N}$ || *NPT* || *LTT* or *NID* || *LKT* (=*KMT* || *LMT*) || $Info_{N}$, to a proxy node *j*.
2. The proxy node *j* selects a location key function according to *LKT*; then the proxy node *j* inputs the Request, *LKP*, *LKS*, and $Info_{PN}$ into the location key function. Thereby, selected operations between the node and proxy node are executed. For instance, the client is able to obtain *PK* if *KMT* is a key-sharing, and so on. The client is also able to obtain *LT* if *LMT* is a location

certification.

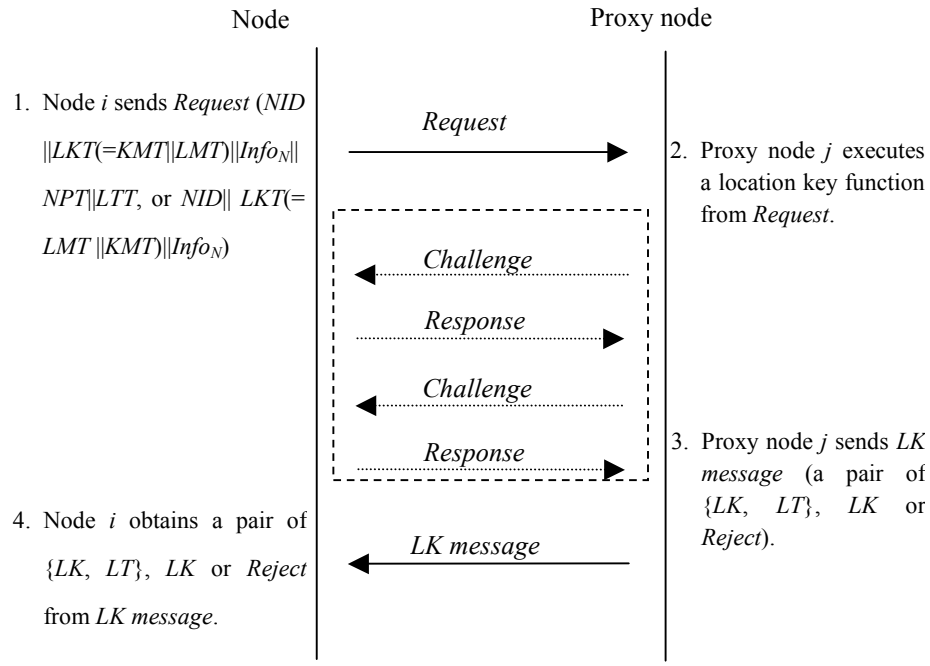


Fig. 2.2.2. Sequence of proposed method

3. The proxy node *j* obtains a pair of {*LK*, *LT*} (if *LKT* is *KMT* || *LMT*), *LK* (if *LKT* is *LMT* || *KMT*) or *Reject*, and then the proxy node *j* sends a part (is chosen by *LKP*) of output of the location key function as a *LK message* to the node *i*.
4. The node *i* receives the *LK message*.

5. 2. 3. 4. Combinations of key management and location management

Table 5.2.3 shows combinations of key management and location management. The study introduces some instances of combinations from Table 5.2.3.

- LK function Combination 2: is a scheme by which a proxy node verifies the node location; then the proxy node shares a symmetric key with the node if the location is within 50 [m] from the proxy node. For example, scheme [BM02] exists.
- KL function Combination 3: is a group key generation in which group members can certificate who, where, and when to share the group key with a third party, using the scheme [AM04]
- LK function Combination 4: is a scheme by which a proxy node verifies the location of a node; then the proxy node distributes a decryption key for encrypted data (e.g. business documents) if the location is in the specific area (e.g. an office). For example, scheme [CN02] exists.
- LK function Combination 6: is a scheme by which a proxy node verifies location of a node and then the proxy node allows the node to use a digital sign key if the location is in the president's

office.

- KL function Combination 9: is a scheme by which all meeting participants share a group key and the proxy node issues a location certificate including the meeting location and the group key.
- KL function Combination 12: is a scheme by which a proxy node revokes a decryption key of a node and verifies the location of the revocation. Subsequently, the proxy node opens a gate for the node if the location is in a specific area.

The study believes that other concrete schemes exist in addition to the above instances.

Table 5.2.3. Combinations of key management and location management

Location / Key	Verification	Certification
Issuing	KL function Combination 1	KL function Combination 7
	LK function Combination 1	LK function Combination 7
Sharing	KL function Combination 2	KL function Combination 8
	LK function Combination 2 [BM02]	LK function Combination 8
Generation	KL function Combination 3	KL function Combination 9
	LK function Combination 3	LK function Combination 9
Distribution	KL function Combination 4	KL function Combination 10
	LK function Combination 4 [CN02]	LK function Combination 10
Revocation	KL function Combination 5	KL function Combination 11
	LK function Combination 5	LK function Combination 11
Access control	KL function Combination 6	KL function Combination 12
	LK function Combination 6	LK function Combination 12

5. 2. 3. 5. Use of Location token

On a KL function, a proxy node gives a location token *LT* as output of a KL function to a node. Plural verification methods (for *LT*) differ according to *LT* types. Therefore, a node should demand an *LT* type that the node can verify or a third party selected by the node can verify. In addition, an *LT* trusted level and an *LT* type depend on a domain of an *LT* issuer. A domain is an attribute that a token issuer belongs. The paper [MIDP2.0] classifies domains into trusted domains (operator, manufacture and trusted third party) and an untrusted domain. Alike the paper, the proposed method sets some different powers into the corresponding domains. Moreover, the proposed method applies the domain concept to MACs except for signatures. A verifier of *LT* can evaluate a trusted level of *LT* using the domain and the type of *LT*. A node can receive cryptographic LBSs from a proxy node

or a third party if the node provides *LT* from another proxy node to the proxy node or the party. A new *LT* must include a domain or a type of the original *LT* if the proxy node issues the new *LT* that the proxy node transforms the original *LT* (e.g. a location reference *LR*) because security of the new *LT* depends on the original *LT*.

5. 2. 4. Evaluation

This section shows that the proposed method satisfies the following four requirements:

5. 2. 4. 1. Availability

By this proposed method, a proxy node has location key functions; a PJ function can control these location key functions using a location key policy. Consequently, only the allowed node can use the functions. This study presumes that a KM function or an LM function (in the location key functions) authenticate node identification.

5. 2. 4. 2. Universality

As shown in Table 5.2.1, the proposed method assumes four types of location tokens that correspond to combinations of four types of proxy nodes and three types of location measurement techniques. A node can request the type of provided location token to the proxy node. Therefore, various existing systems can adopt the proposed method. Moreover, existing key and location management schemes can apply to a KM function and an LM function of the proposed method because the proposed method treats the KM function and LM function as nodes.

5. 2. 4. 3. Associativity

As shown in Figure 5.2.1, a KL function structure can force output of a KM function to be input of an LM function for boosting the relation between the KM function and LM function. In the same way, an LK function structure can force output of an LM function to be input of a KM function.

On location key functions, a KM function and an LM function can authenticate the same node using a *CC* value, for preventing differences between nodes that the KM function and the LM function authenticate. Here, a *CC* value (a client ID *CID*, a random number *R*, a processed key *PK*, and a location token) depends on a client by which each management function targets as follows:

1. A server can identify a client, because *CID* is a unique ID.
2. A server can distinguish a client that has *R* from another client that does not have the *R*. But the server cannot identify a client, because the *R* is a temporal ID by which the server gives for providing LBSs.
3. A server can distinguish a client that has *PK* from another client that does not have the *PK*. But the server may not be able to identify a client, because a key management scheme may not

authenticate the client on a KM function.

4. A server can distinguish a client that has *LT* from another client that does not have the *LT*. But the server may not be able to identify a client, because the server may not identify the client for location management.

Thus, a sever can authenticate the same client using a *CC* value when the client requests anonymity on client authentication, and when management methods of an LM function and a KM function do not support a temporal ID. Here, securities of *CC* values rely on securities of the management methods.

In addition, location key functions inputs are limited to a Request and Responses. A PJ function can verify the Request directly. On the other hand, a KM function and an LM function verify the Responses; a PJ function verifies feedback from the KM function and an LM function. In a word, the PJ function can verify the Responses indirectly. The PJ function can also stop execution of an LK function if the PJ function receives feedback that the KM function and LM function are unable to authenticate the same node. Therefore, the PJ function can verify a management path between the KM function and LM function.

Consequently, location key functions can satisfy *Associativity* using 1) a context connection value; 2) a construct that output of a management function inputs another management function; and 3) policy-based access control.

5. 2. 4. 4. Privacy

A proxy node outputs only location token *LT* as privacy information of a node. A node can demand anonymity, location hiding and time hiding to the proxy node using a requested node privacy type *NPT*. On the other hand, the proxy node generates a location token, which is excluded privacy information selected by the *NPT*, if a PJ function allows generation of *LT* from a location key policy *LKP*.

5. 2. 4. 5. Security Analysis

The study considers security of the proposed method. From Section 5.2.2.5 and Section 5.2.4.3, location key functions are secure for external attacks. In addition, a location key server is a trusted party when the server is a station. Consequently, the proposed method can prevent attacks described in Section 5.2.2.5 if a location key server type is a station. The security of the proposed method is equal to trustiness of a location key server when the server type is a mobile, a sensor, or a tag.

5. 2. 5. Conclusion

This paper proposed a method of constructing secure cryptographic LBSs, which have a location key function consisting mainly of a location management function and key management function. This

proposed method includes three approaches: apply a construct by which output of a management function inputs another management function, context connection value, and policy-based access control to location-key functions. In addition, the thesis suggested new cryptographic LBSs covering all combinations of key management and location management.

Chapter 6. Conclusion and Future Works

6. 1. Conclusion

This thesis defined a system model including plural properties on targeted wireless mobile networks (e.g. ad-hoc / mesh networks, cellular phone / wireless LAN spot services, and ubiquitous computing / networks). The principal property is that a mobile node is not always possible to connect to a station, which is a gate of a backbone infrastructure. For example, stations are base stations of cellular phone services, and access points of wireless LAN spot services.

The principal property causes three security problems of existing key management technologies. For solving the problems, the thesis assumed a "*Proxy Node*" instead of a station on the system model. The proxy node is temporally authorized by stations, or is elected by general nodes. Existing location management technologies also have two security problems on the system model.

Two of the above five problems are new problems by which the thesis indicates for the first time. For the new two problems, the thesis showed new two concepts: 1) "*Interaction Key*" in a key management technology, and 2) "*Plural Provers Verifiable Location Verification*" in a location management technology.

The thesis proposed three new key management schemes and two new location management schemes as solutions of the above five problems. Moreover the thesis showed validity of the five proposed schemes.

In addition, the thesis considered security of cryptographic location-based services (LBSs) that consist mainly of key management technologies and location management technologies, and then proposed a new method of constructing secure cryptographic LBSs. The thesis also suggested a potential for new cryptographic LBSs by showing plural combinations of key management and location management functions.

6. 2. Future Works

This subsection shows three future works of this thesis as follows:

1. for realizing the proposed location verification schemes (described in Chapter 4),
2. for verifying validity of the proposed secure cryptographic LBSs constructing method (described in Chapter 5) by combining existing key management and location management functions, and
3. for proposing secure cryptographic LBSs that integrate into the proposed key management functions (described in Chapter 3) and location functions (described in Chapter 4).

List of Papers

Papers Published in Journals

[AM04-1] J. Anzai and T. Matsumoto, "Interaction Key Generation Schemes," IEICE Trans. Fundamentals. Vol. E87-A, No. 1, pp. 152-159, January 2004.

[AM04-2] J. Anzai and T. Matsumoto, "Integration of the Incentive and the PKI-supporting Mechanism on Multi-hop Cellular Networks," In IPSJ Journal, Vol. 45, No. 12, pp. 2589-2599, December 2004. (in Japanese)

[AM05-3] J. Anzai and T. Matsumoto, "A Distributed User Revocation Scheme for Ad-Hoc Networks," IEICE Trans. Communications., to appear in September 2005 (accepted in April 2005).

[AM05-4] J. Anzai and T. Matsumoto, "Location Verification Schemes Resistant Against Relay Attack," submitted to IEICE Trans. Fundamentals., (submitted in June 2005).

[AM05-5] J. Anzai and T. Matsumoto, "Plural Provers Verifiable Location Verification Schemes," submitted to IEICE Trans. Fundamentals., (submitted in June 2005).

International Conference Proceedings

[AM05-6] J. Anzai and T. Matsumoto, "How to Construct Secure Cryptographic Location Based Services," SECUBIQ 2005 --- International Workshop on Security Ubiquitous Computing Systems, ., to appear in December 2005 (accepted in August 2005).

Technical Reports

[AM03] J. Anzai, T. Matsumoto, "Interaction Key and Its Generation," In Proceedings of 2003 Symposium on Cryptography and Information Security, 4B-4, January 2003. (in Japanese)

[AM04] J. Anzai, T. Matsumoto, "The incentive and PKI-supporting mechanisms for general multi-hop cellular networks," In Proceedings of 2004 Symposium on Cryptography and Information Security, 2C2-4, January 2004. (in Japanese)

[AM05-1] J. Anzai, T. Matsumoto, "Location Verification (1): Location Verification Schemes Resistant Against Relay Attack," In Proceedings of 2005 Symposium on Cryptography and Information Security, 2B4-3, January 2005. (in Japanese)

[AM05-2] J. Anzai, T. Matsumoto, "Location Verification (2): Plural Provers Verifiable Location Verification Schemes," In Proceedings of 2005 Symposium on Cryptography and Information Security, 2B4-4, January 2005. (in Japanese)

[AM05-3] J. Anzai, T. Matsumoto, “A Construction Method of Secure Cryptographic Location-Based Services,” In Technical report of IEICE, ISEC2005, September 2005. (in Japanese)

Other Related Papers

[AMM01] J. Anzai, N. Matsuzaki, and T. Matsumoto, “A Flexible Method for Masked Sharing of Group Keys,” IEICE Trans. Fundamentals, Vol. E84-A, No.1, pp. 239-246, January 2001.

[AM05-7] J. Anzai and T. Matsumoto, “A Consideration on the Location Certification Infrastructure of Physical Objects,” submitted to IPSJ Journal, (submitted in May 2005).

Other Related International Conference Proceedings

[AMM99-4] J. Anzai, N. Matsuzaki and T. Mastumoto, “A quick group key distribution scheme with "entity revocation",” Advances in Cryptology --- ASIACRYPT'99, LNCS1716, pp. 333-347, Springer-Verlag, November 1999.

Other Related Technical Reports

[AM99-1] J. Anzai, N. Matsuzaki and T. Matsumoto, “A Method for Masked Sharing of Group Keys,” In Proceedings of 1999 Symposium on Cryptography and Information Security, F1-3.2, January 1999. (in Japanese)

[AM99-2] J. Anzai, N. Matsuzaki and T. Matsumoto, “A Method for Masked Sharing of Group Keys (2),” In Technical report of IEICE, ISEC98-81, March 1999. (in Japanese)

[AM99-3] J. Anzai, N. Matsuzaki and T. Matsumoto, “A Method for Masked Sharing of Group Keys (3),” In Technical report of IEICE, ISEC99-38, September 1999. (in Japanese)

Bibliography

[AD02] E. R. Anton and O. C. M. B. Duarte, “Group Key Establishment in Wireless Ad Hoc Networks,” In Proc. of Workshop em Qualidade de Servico e Mobilidade - 2002, 2002.

[AODV] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc On-Demand Distance Vector (AODV) Routing,” RFC3561, IETF Network Working Group, 2003.

[BB02] S. Buchegger and J. Y. L. Boudec, “Performance analysis of the CONFIDANT protocol: Cooperation of nodes-fairness in dynamic ad-hoc networks,” In Proc. of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, IEEE, 2002.

[BC93] S. Brands, D. Chaum, “Distance-Bounding Protocols,” In Proc. of Eurocrypto’93, Springer-Verlag, pp. 344-359, 1993.

[BD94] M. Burmester and Y. Desmedt, “A Secure and Efficient Conference Key Distribution System,” In Proc. of EUROCRYPT'94, LNCS950, pp. 275-285, Springer-Verlag, 1994.

[BF97] D. Boneh, M. Franklin, “Efficient Generation of Shared RSA Keys,” In Proc. of CRYPTO'97, pp. 425-439, Springer-Verlag, 1997.

[BH00] L. Buttyan and J. P. Hubaux, “Enforcing service availability in mobile ad-hoc WANs,” In Proc. of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, 2000.

[BH02] L. Buttyan and J. P. Hubaux, “Stimulating cooperation in self-organizing mobile ad hoc networks,” ACM Journal for Mobile Networks, special issue on Mobile Ad Hoc Networks, 2002.

[BM02] S. Banerjee, A. Mishra, “Secure Spaces: Location-based Secure Wireless Group Communication,” Mobile Computing and Communications Review, Volume 1, Number 2, 2002.

[BP00] P. Bahl and V. N. Padmanabhan, “RADAR: An In-Building RF-Based User Location and Tracking System,” In Proc. of IEEE infocom 2000, pp. 775.784, 2000.

[CBH02] S. Capkun, L. Buttyan, and J. P. Hubaux, “Self-Organized Public-Key Management for Mobile Ad-Hoc Networks-Abstract,” in Report on a Working Session on Security in Wireless Ad Hoc Networks, ACM Mobile Computing and Communications Review (MC2R), vol. 6, no. 4, 2002.

[CBH03] S. Capkun, L. Buttyan, and J. P. Hubaux, “SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks,” In Proc. of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03), 2003.

[CH91] D. Chaum, E. van Heyst, “Group signatures,” In Proc. of EUROCRYPT'91, LNCS547, pp. 257-265, Springer-Verlag, 1991.

[CH04] S. Capkun, J. P. Hubaux, “Securing position and distance verification in wireless networks,” Technical report EPFL/IC/200443, submitted to ACM MobiCom04, 2004.

[CHJ04] S. Capkun, J. P. Hubaux, and M. Jakobsson, “Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks,” EPFL-IC Technical report IC/2004/10, 2004.

[CK00] S. Cho and C. Kim, “A Secure Multicast Architecture with the Decentralized Key Management,” In Proc. of International Conference on Electronic Commerce 2000, 2000.

[CN02] M. D. Corner, B. D. Noble, “Zero-Interaction Authentication,” In Proc. of MOBICOM'02, 2002.

[DF89] Y. Desmedt, Y. Frankel, “Threshold Cryptosystems,” In Proc. of CRYPTO'89, LNCS435, pp. 307-315, Springer-Verlag, 1989.

[DH76] W. Diffie, M. E. Hellman, “New directions in cryptography,” IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, 1976.

[DHS03] V. Daza, J. Herranz, and G. Saez, “Constructing General Dynamic Group Key Distribution Schemes with Decentralized User Join,” In Proc. of ACISP'03, pp. 464-475, Lecture Notes in Computer Science, Springer-Verlag, 2003.

[DSR] David B. Johnson, David A. Maltz, and Yih-Chun Hu, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR),” ITERNET-DRAFT, draft-ietf-manet- dsr-09.txt, IETF MANET Working Group, April 2003.

[FA04] K. B. Frikken, M. J. Atallah, “Privacy Preserving Route Planning,” In Proc. of ACM Workshop on Privacy in the Electronic Society(WPES'04), 2004.

[G99] N. Gilboa, “Two Party RSA Key Generation,” In Proc. of CRYPT0'99, pp. 116-129, Springer-Verlag, 1999.

[GW98] E. Gabber and A. Wool. “How to prove where you are: Tracking the location of customer equipment,” In Proc. of 5th ACM Conf. Computer and Communications Security (CCS), pp. 142-149, 1998.

[HBC01] J. P. Hubaux, L. Buttyan, and S. Capkun, “The Quest for Security in Mobile Ad Hoc Networks,” In Proc. of the ACM Symposium on Mobile Ad Hoc Networking and Computing, 2001.

[HMYMMA04] K. Hirasawa, M. Minami, S. Yokoyama, M. Mizumachi, H. Morikawa and T. Aoyama, “Implementation and Evaluation of a Distributed Ultrasonic Positioning System,” TECHNICAL REPORT OF IPSJ, 2004-MBL-029, 2004. (in Japanese)

[HPJ03-1] Y. C. Hu, A. Perrig, and D. B. Johnson, “Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols,” In Proc. of 2nd ACM Workshop on Wireless Security, 2003.

[HPJ03-2] Yih-Chun Hu, A. Perring, and D. B. Johnson, “Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks,” In Proc. of the Twenty-Second Annual Joint Conference of the IEEE (INFOCOM ‘03), 2003.

[HS90] S. Haber, W. S. Stornetta, “How to Time-Stamp a Digital Document,” In Proc. of CRYPTO'90, pp. 437-455, Springer-Verlag, 1990.

[ITWUSM00] M. Izumi, S. Takeuchi, Y. Watanabe, K. Uehara, H. Sunahara, J. Murai, “A Proposal on a Privacy Control Method for Geographical Location Information Systems,” In Proc. of INET’00, 2000.

[J04] A. Juels, “Yoking-Proofs” for RFID tags, In Proc. of First International Workshop on Pervasive Computing and Communication Security, IEEE Press, 2004.

[JHB03] M. Jakobsson, J. P. Hubaux, and L. Buttyan, “A Micro- Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks,” In Proc. of the Seventh International Financial Cryptography Conference, 2003.

[K96] P. Kocher, “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems,” In Proc. of CRYPTO'96, Springer-Verlag, pp. 104-113, 1996.

[KAM01] T. Kato, J. Anzai and N. Matsuzaki, “Double Exponentiations Accelerator with Mode Switching for Mobile Terminals,” In Proc. of WPMC2001, 2001.

[KD98] K. Kurosawa and Y. Desmedt, “Optimum Traitor Tracing and Asymmetric Scheme,” Advances in Cryptology-EUROCRYPT'98, pp. 145-157, Springer-Verlag, 1998.

[KIAM00] T. Kato, S. Ito, J. Anzai, N. Mastuzaki, “A Design for Modular Exponentiation coprocessor in Mobile Telecomm-unication Terminals,” In Proc. of CHES2000, pp. 216-228, 2000.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun, “Differential Power Analysis,” In Proc. of Crypto'99, Springer-Verlag, pp. 388-397, 1999.

[KMSW01] H. Kurnio, L. McAven, R. Safavi-Naini and H. Wang, “Efficient Revocation Schemes for Secure Multicast,” In Proc. of ICISC'01, pp. 160-177, 2001.

[KMSW02-1] H. Kurnio, L. McAven, R. Safavi-Naini, and H. Wang, “A dynamic group key distribution scheme with flexible user join,” In Proc. of ICISC'02, LNCS2587, Springer-Verlag, pp. 478-496, 2002.

[KMSW02-2] H. Kurnio, L. McAven, R. Safavi-Naini, and H. Wang, “A Group Key Distribution Scheme with Decentralized User Join,” In Proc. of the 3rd Conference on Security in Communication Networks - SCN2002, 2002.

[KNF02] T. Kitasuka, T. Nakanishi, and A. Fukuda, “Indoor Location Sensing Techinique using Wireless Network,” In Proc. of Computer System Symposium’02, pp. 83-90, 2002. (in Japanese)

[KNF03] T. KITASUKA, T. NAKANISHI and A. FUKUDA, “Indoor Location Sensing Technique Using Wireless Network,” In IPSJ Journal, Vol. 44, No. SIG 10(ACS 2), pp. 131-140, July 2003. (in Japanese)

[KNF04] T. Kitasuka, T. Nakanishi and A. Fukuda, “An Implementation of Wireless LAN based Indoor Positioning System WiPS,” In Proc. of DICIMO2004, pp. 349-352, July 2004. (in Japanese)

[KZLLZ01] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing Robust and Ubiquitous

Security Support for Mobile Ad Hoc Networks," In Proc. of the 9th International Conference on Network Protocols (ICNP), 2001.

[MAMGA00] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP," RFC2560, IETF Network Working Group, 1999.

[MGLB00] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," In Proc. of The Sixth International Conference on Mobile Computing and Networking 2000, 2000.

[MIDP2.0] JSR 118 Expert Group, Java Community Process, "Mobile Information Device Profile for Java 2 Micro Edition Version 2.0," 2002.

[MK03] A. MACHIZAWA and Y. KITAGUCHI, "An Analysis of Software time-stamping accuracy by Super Precision Time-Stamper, " TECHNICAL REPORT OF IEICE, Vol. NS2003, No. 159 pp.63-66, 2003. (in Japanese)

[NNT03] K. Nakanishi, J. Nakazawa, and H. Tokuda, "LEXP: Preserving User Privacy and Certifying the Location Information," In Proc. of 2nd Workshop on Security in Ubiquitous Computing Ubicomp'03, 2003.

[NP00] M. Naor and B. Pinkas, "Efficient Trace and Revoke Schemes," In Proc. of Financial Cryptography2000, Lecture Notes in Computer Science, Springer-Verlag, 2000.

[NW04] M. Numao and Y. Watanabe, "Dynamic Group Key Construction for P2P Multicast," IPSJ Journal, vol. 45, no. 2, pp. 597-604, 2004. (in Japanese)

[OTWFYSK03] A. Ogino, K. Tsunehara, K. Watanabe, K. Fujishima, R. Yamasaki, H. Suzuki and T. Kato, "Integrated Wireless LAN Access System - Study on Location Method - ," In Proc. of DICIMO2003, June 2003. (in Japanese)

[P91] T. P. Pedersen, "A Threshold Cryptosystem without a Trusted Party," In Proc. of EUROCRYPT'91, pp. 522-526, Springer-Verlag, 1991.

[PS98] G. Poupard, J. Stern, "Generation of Shared RSA Keys by Two Parties," In Proc. of ASIACRYPT'98, pp.11-24, Springer-Verlag, 1998.

[RST01] R. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," In Proc. of ASIACRYPT2001, pp. 552-565, Springer-Verlag, 2001.

[S79] A. Shamir, "How to Share a Secret," Comm. Assoc. Comput. Mach., vol. 22, no. 11, pp. 612-613, 1979.

[SBHJ03] N. B. Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson, "A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular NetWorks," In Proc. of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing, 2003.

[SS04] J. Saito, K. Sakurai "Grouping proof for RFID tags," In Proc. of the 2004 Computer Security Symposium (CSS2004), 2B-1, 2004. (in Japanese)

[SSL] A. Freier, P. Karlton, and P. Kocher, “The SSL Protocol Version 3.0,” Internet-Draft, 1996.

[SSW03] N. Sastry, U. Shankar, and D. Wagner, “Secure Verification of Location Claims, ” Report No. UCB//CDS-03-1245, University of California, Berkekey.

[STW00] M. Steiner, G. Tsudik, and M. I. Waidner, “Key agreement in dynamic peer groups,” IEEE Transactions on Parallel and Distributed Systems, vol. 11, no. 8, pp. 769-780, 2000.

[TMIK05] H. Toriyama, A. Machizawa, T. Iwama and A. Kaneko, “Development of A Hardware SNTP Server,” TECHNICAL REPORT OF IEICE, 2005-04-CQ-RCS , 2005.

[TLS] T. Dierks, C. Allen, “The TLS Protocol Version 1.0,” Request for Comments:2246, 1999.

[TO98] Y. Tamura, E. Okamoto, “Concept and Implementation of Flexible Secret Sharing Scheme,” In Proc. of Computer Security Symposium'98, pp. 21-26, 1998. (in Japanese)

[VN04] A. Vora, M. Nesterenko, “Secure Location Verification Using Radio Broadcast,” In Proc. of OPODIS 2004: 8th International Conference on Principles of Distributed Systems, 2004.

[WF03] B. R. Waters, E. W. Felten, “Secure, Private Proofs of Location,” Princeton University Computer Science Technical Reports, TR-667-03, 2003.

[WGL97] C. Wong, M. Gouda, and S. Lam, “Secure Group Communications Using Key Graphs,” In Proc. of ACM SIGCOMM'98, 1998. Also, Technical Report TR 97-23, Department of Computer Sciences, The University of Texas at Austin, July 1997.

[WHA99] D. Wallner, E. Harder, and R. Agee, “Key Management for Multicast: Issues and Architectures,” Internet RFC2627, June 1999.

[WKSEYY03] H. Watanabe, T. Kato, R. Sasaki, Y. Eguchi, Y. Yasunaga, and K. Yoshida, “Evaluation on Dynamic Group Key Generation Methods under P2P Environment,” IPSJ Journal, vol. 44, no. 8, pp. 2155-2162, 2003. (in Japanese)

[WT01] A. Weimerskirch and G. Thonet, “A Distributed Light-Weight Authentication Model for Ad-hoc Networks,” In Proc. of Information Security and Cryptology-ICISC2001, 2001.

[WTTUM96] Y. Watanabe, S. Takeuchi, F. Teraoka, K. Uehara, and J. Murai, “The Geographical Location Information System with Privacy Protection,” IPSJ Journal, Vol. 37, No. 6, 1996. (in Japanese)

[YK02] S. Yi and R. Kravets, “Key Management for Heterogeneous Ad Hoc Wireless Networks,” Technical Report UIUCDCS-R-2002-2290/UILU-ENG-2002-1734, University of Illinois at Urbana-Champaign, 2002.

[ZH99] L. Zhou and Z. Haas, “Securing Ad Hoc Networks,” IEEE Network, vol. 13, no. 6, pp. 24-30, 1999.

[ZYC02] S. Zhong, Y. R. Yang, and J. Chen, “Sprite: A Simple, Cheat-proof, Credit-based System for Mobile Ad Hoc Networks,” Technical Report Yale/DCS/TR1235, Department of Computer Science, Yale University, 2002.